MERIT Master Thesis

TITLE: Design and implementation of a loudness monitoring system

AUTHOR: Jordi Cenzano Ferret

DIRECTOR: Josep R. Casas Pla

DATE: 17-07-2013

Index

1 Introduction 5

2 Loudness motivation 7

3 Background and references 8

3.1 Human auditory system 8

3.2 Audio: Sound into electrical world 10

3.2.1 Audio impedance 11

3.2.2 Analog level: dBu 11

3.2.3 Digital level: dB FS 12

3.2.4 Audio level references 13

3.2.5 Dynamic range (sound and audio) 14

4 Audio level meters 16

4.1 Short term audio level meters 16

4.1.1 Peak Program meter (or PPM) 16

4.1.2 Volume Unit Meter (VU-Meter) 17

4.1.3 True peak meter 17

4.2 Long term audio level meters (Loudness meters) 19

4.2.1 Zwicker loudness method 19

4.2.2 Equivalent level (Leq) 20

4.2.3 ITU-R BS.1770 22

4.2.4 Dobly AC-3 and Dialnorm 23

4.3 Other audio meters 25

4.3.1 Spectrum analyzers 25

4.3.2 Phase meters 25

5 Loudness standards 26

5.1 EBU Recommendation 128 26

5.2 Recommended practice A/85 28

5.3 ARIB TR-B32 29

6 State of the art of loudness meters 30

6.1 Standalone loudness meters 30

6.2 Software loudness meters 32

6.3 Audio mixers with loudness meters 33

7 Our loudness meter implementation 35

7.1 Loudness Measuring Engine (LME) 36

7.1.1 Performance 41

7.1.2 “EBU mode” compliance 43

7.2 Directshow wrapper 44

8 Our Loudness monitoring system (LMS) 46

8.1 LMS introduction 46

8.2 LMS Architecture 48

8.3 Implementation issues 49

8.3.1 MySQL C++ Connector BUG 49

8.3.2 LMS time synch 49

8.3.3 Microsoft Broadcast Driver Architecture stability 50
8.4 LMS core modules 51
8.4.1 Log to DB Module 52
8.4.2 DTT Module 52
8.4.3 CALC Module 54
8.4.4 SHOW Module 56
8.4.5 CLEAN Module 58
8.5 LMS satellite modules 60
8.5.1 Playout change over controller module 60
8.5.2 ASP SHOW Module 61
9 Example measures 62
9.1 8tv 63
9.2 RAC105tv 65
9.3 TV3 67
9.4 Comparative of channel loudness 69
9.4.1 Per channel 69
9.4.2 Per channel and type of content 70
9.4.3 Histogram and LRA 71
10 Future work 73
11 Reviewing objectives 74
12 Conclusions 75
13 Acronyms table 76
14 References 77
15 Annexes 80
15.1 Manual of command line Loudness meter 81
15.1.1 Introduction 82
15.1.2 Input modifiers 82
15.1.3 Loudness configuration file 83
15.1.4 Accepted input wave (.wav) formats 85
15.1.5 Output wave format 86
15.2 Measuring engine performance raw data 87
15.3 “EBU Mode” compliance test results 90
15.4 Manual of CJOCLoudnessDS directshow filter 92
15.4.1 Introduction 93
15.4.2 Input audio data 93
15.4.3 Filter configuration parameters 93
15.5 LMS Database description 96
15.5.1 Schema 97
15.5.2 Tables 97
15.5.2.1 Channels 97
15.5.2.2 ChannelType 98
15.5.2.3 Asrun 98
15.5.2.4 itemLoudness 98
15.5.2.5 Items 99
15.5.2.6 ItemType 99
15.5.2.7 Captures 99
15.5.2.8 Samples 100
15.5.2.9 Configdata 100
15.5.3 SQL creation code 101

15.6 Manual of LMS core modules....105
15.6.1 Manual of log to DB LMS core module....106
15.6.2 Compliant asrun file syntax....107
15.6.3 Manual of DTT LMS core module....108
15.6.4 Manual of CALC LMS core module....109
15.6.5 Manual of SHOW LMS core module....110
15.6.5.1 Report by broadcasted event....110
15.6.5.2 Report by channel capture....111
15.6.5.3 Report by channel dates....112
15.6.5.4 Report by loudness level....113
15.6.6 Manual of CLEAN LMS core module....115
15.7 Loudness graphs....116
15.7.1 Understanding loudness graphs....117
15.7.2 Broadcasted events in loudness graphs....118
15.7.3 Zooming in loudness graphs....119
15.7.4 Export loudness graph data....119

1 Introduction

This document reports on a project on a Loudness Monitoring System designed and implemented by the author at a broadcaster premises.

The main objective of the project is to **explore the normalization of audio levels in the media industry**, which is a relevant issue nowadays. Many broadcast organizations around the world are concerned about this problem and they have published many papers and standards that will be analyzed in this project, some of those organizations are: European Broadcasting Union (EBU), International Telecommunication Union (ITU), Advanced Television Standards Committee (ATSC), etc.
I want to notice that also in Catalonia exists a workgroup organized by Consell del Audiovisual de Catalunya (CAC) working in loudness normalization. The Universitat Politecnica de Catalunya(UPC) and the author are members of this workgroup.

Specific objectives of the project are:

- **Study and summarize background, references, and state of the art (competing) systems.**

- **Implement a Loudness meter following the last published standards.**

- **Evaluate the performance of the implemented loudness meter.**

- **Design and implement a prototype of a complete loudness monitoring system based in our loudness meter.**

This report is structured in three large sections as described as follows:

- The first section of this project introduces some sound and audio concepts, and it presents the different standards and meters involved in loudness metering: In chapter 2 we explain why the loudness normalization is needed, and why now and not 20 years ago. Next, in chapter 3 basic sound and audio concepts are introduced and loudness is defined.
  In chapter 4 different kinds of audio meters are presented (short term level meters, long term level meters, and some others).
  The most used loudness standards around the world are explained in chapter 5.
  Finally the state of the art of loudness meters is presented in chapter 6.

- The second section of this thesis is focused on our loudness system implementation: Chapter 7 explains the implementation details of our loudness meter. Then, in chapter 8 we explain how we built a complete loudness monitoring system based in the loudness meter presented previously.
  Once we have constructed the complete loudness monitoring system we evaluate it measuring loudness in a real broadcast environment (chapter 9).

- The third section announce the future work, review the proposed objectives, and the conclusions obtained doing this project will be presented.

SECTION 1: Loudness motivation and background

2 Loudness motivation

In recent years the broadcasters and regulatory administrations from around the world have received many complaints of the audience about the audio level shifting between channels, or between programs in the same channel. This usually happens between programs and advertisements. The audio level differences annoy the audience, and force them to get the remote control and adjust the volume to their comfort level many times per day.

The TV and radio systems exists from a many years ago, but the issue of audio level differences between programs has been noticeable in the recent years coinciding with the digitizing of its systems. We thing that the reason of this could be a mix of the following reasons:

- With the digitizing, the systems involved in the sound masterization of media products have become more complex and powerful, and the sound engineers can use this new tools to maximize the presence (or loudness) of their products.

- The quality of audio user systems increased a lot (stereo, home theater, Dolby digital 5.1, etc…) and the user becomes more exigent.

- The number of channels that the user can reach has been increased, and nowadays exist smaller TV stations that because its budgets cannot be as concerned with signal quality as the biggest ones.

Each country has reacted in a different way to these audio level shifting complains, for instance USA has published the “Commercial Advertisements Loudness Mitigation Act” (CALM) law [1] that take effect on December 13th of 2012. This law is requiring broadcast and cable television stations to adopt industry technology that ensures commercials are not louder than regular programming.
The EBU has published the EBU R128 recommendation [2]. This recommendation has been adopted in a different way by various European Union (EU) country members, France as a law, Germany as a directive, Spain as a recommendation, etc…

3 Background and references

In the first part of this chapter introduces some concepts related with sound and audio that are needed to understand the following sections. Then the most common audio meters are presented (peak meter, VU-meter, etc…), and finally the state of the art of loudness meters are explained in detail.

3.1 Human auditory system

The human auditory system (Figure 3-1) is a complex system that transforms atmospheric pressure changes into information that can be interpreted by our brain.

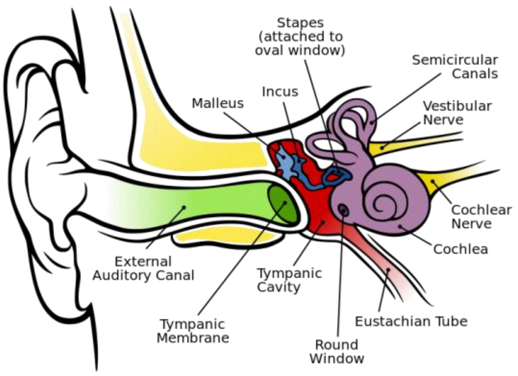


Figure 3-1: **Human audition system (Picture from *[3]*)**

The following figure shows how a fast atmospheric pressure variation becomes an intelligible sound. The unit to measure the sound pressure is pascal (Pa), which is defined as follows:

$$1\ Pa = 1\frac{N}{m^2} = 1\frac{1Kg}{m * s^2} \qquad \textit{3-1}$$

Where: **N** = newton, **m** = meter, **Kg** = Kilogram, **s** = second

1- Silence
2- Audible sound
3- Atmospheric pressure reference level
4- Sound pressure

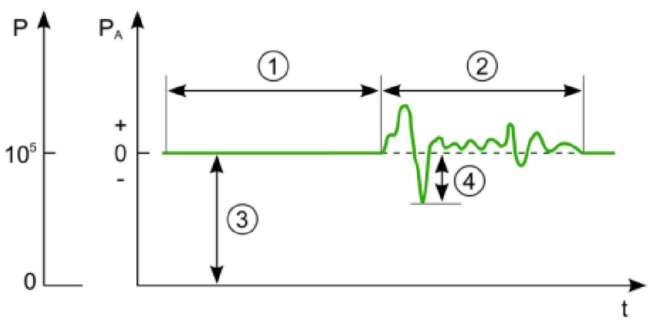


Figure 3-2: Sound and atmospheric pressure (Picture from [4])

Loudness is then defined as follows:

"Attribute of auditory sensation in terms of which sounds can be ordered on a scale extending from quiet to loud" [5]

From this definition we can deduce that loudness is a subjective measure that depends on different factors such as: frequency, bandwidth, direction, duration.

In 1933 Harvey Fletcher and Wilden A. Munson published the Fletcher-Munson curves [6] (or equal loudness contours), these curves are a measure of sound pressure over the frequency spectrum, for which a listener perceives a constant loudness **when presented with pure tones** (sine-wave signal). These curves are improved by ISO 226:2003 [7], see Figure 3-3.

To understand the equal loudness contours of Figure 3-3, first we have to define the root mean square value for a set of values as:

$$RMSn = \sqrt{\frac{1}{n}(x_1^2 + x_2^2 + \cdots + x_{1n}^2} \qquad \textit{3-2}$$

And then the root mean square value for a continuous function RMSf as:

$$RMSf = \sqrt{\frac{1}{T_2 - T_1}\int_{T_1}^{T_2} f(t)^2 dt} \qquad \textit{3-3}$$

From equation *3-3* it is easy to calculate the root mean square value for a tone (sin function):

$$RMSsin = \frac{a}{\sqrt{2}} \qquad \textit{3-4}$$

Where: **a** = amplitude of the tone (peak value)

And finally we can define the sound pressure level in dB SPL as:

$$SPL\ [dB\ SPL] = 10Log_{10}\left(\frac{Prms^2}{Pref^2}\right) = 20Log_{10}\left(\frac{Prms}{Pref}\right) \qquad \textit{3-5}$$

Where: **Prms** = Sound pressure in Pa (RMS), **Pref =** 20μPa (RMS)

Definition of phon: ***Is a unit of loudness level <u>for pure tones</u>, 1 phon is equal to 1 dB SPL at a frequency of 1 KHz.***

Joining the phon definition and the SPL definition:

$$1phon = 1\ db\ SPL\ \ @1KHz \qquad 1dB\ SPL\ \ @1KHz\ =\ 20Log_{10}\left(\frac{Prms}{20\mu Pa}\right) \qquad \textit{3-6}$$

We can compute in an easy way that the sound pressure (Prms) associated to 1 phon is:

22,44μPa (@1KHz)

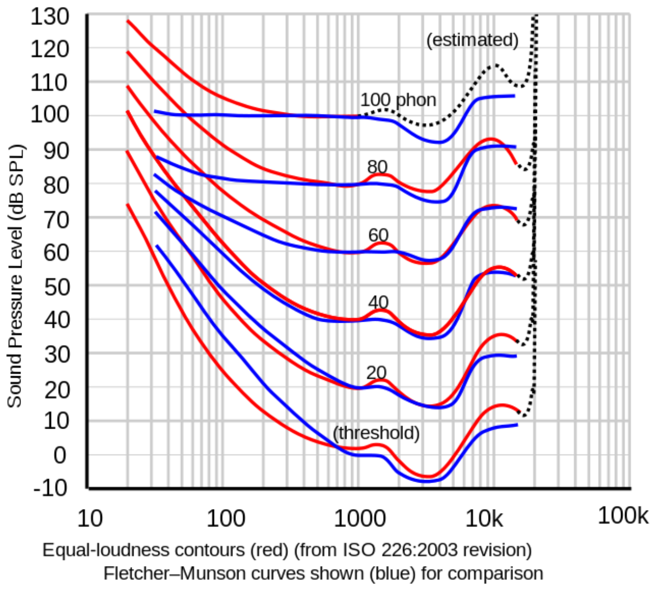


Figure 3-3: Equal loudness contours **(Picture from *[8]*)**

These equal-loudness contours demonstrate that the loudness perception varies depending on frequency and intensity of the sounds. For instance, reading the previous contours we can realize that we have to amplify 70dB a tone of 20Hz to equal the same loudness that induce a tone of 1000Hz of 20phons (from 20dBSPL to 90dBSPL).
And we can see that, at higher pressure levels the curve is flatter than at lower pressure, this means that the variation of loudness perception versus frequency is lower at high pressure levels.

3.2 Audio: Sound into electrical world

When the acoustic air pressure is transformed into electrical signals using a transducer, usually a microphone, there are a large number of measures that can be made to this audio signal. See Figure 3-4.

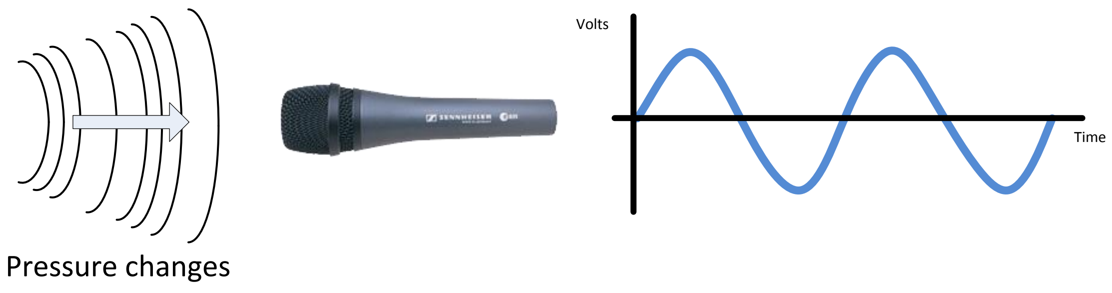


Figure 3-4: From acoustic pressure to electrical world

These measures can be related with: peak values, mean value, frequency, duration, etc…

To properly understand the meaning of the audio measures it is necessary to introduce briefly some audio definitions and concepts.

3.2.1 Audio impedance

About impedances (*), it is important to know that professional audio equipment use a low impedance output drives, and a high impedance input, this is named impedance bridging, see Figure 3-5.

(*) Cables between line output and line input are generally extremely short compared to the audio signal wavelength in the cable, transmission line effects can be disregarded and impedance matching need not be used.

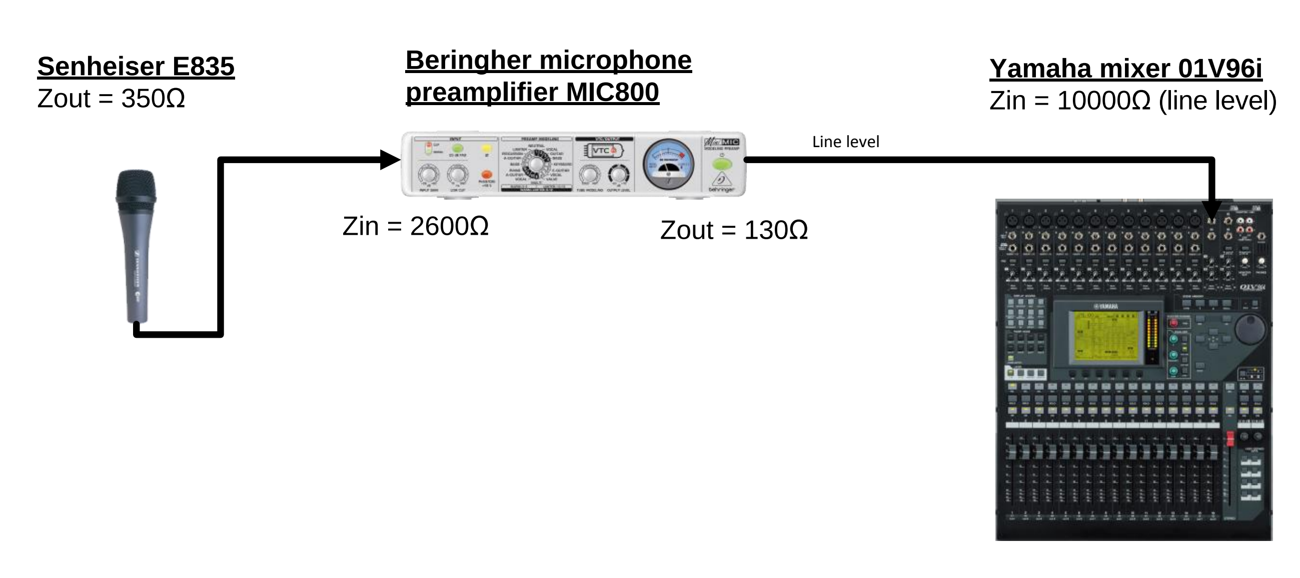


Figure 3-5: Impedance bridging

The loss of signal caused by impedances is defined as follows:

$$Load\ Loss\ [dB] = 20Log_{10}\ \frac{Z_{load}}{Z_{load} + Z_{source}} \qquad \textit{3-7}$$

From equation *3-7* is easy to see that:

If $Z_{load} >> Z_{source} \rightarrow$ Load Loss $\approx$ 0 dB

This means that if you use impedance bridging you do not have to take into account the Load loss.

3.2.2 Analog level: dBu

In order to measure the energy of audio signals the logarithmic unit dBu was created and defined as follows:

$$1\ dBu = 10Log_{10}\ \frac{Vrms^2}{Vref^2} \qquad \textit{3-8}$$

Where **Vrms** = **Vref** = 0.775V (RMS)

We can quantify any signal into dBu using equation *3-8*:

$$A\ dBu = 20Log_{10}\ \frac{Vrms}{Vref} \qquad \textit{3-9}$$

Where **Vrms** = RMS voltage to transform to dBu, and **Vref** = 0.775V (RMS)

Using the dBu definition (*3-8*) and the equation *3-4* we can deduce the audio level in dBu for pure tones (*PTL*) is:

$$PTL\ [dBu] = 20Log_{10}\ \frac{a}{\sqrt{2} \cdot Vref} \qquad \text{3-10}$$

Where: **a** = amplitude of the tone (peak value), **Vref** = 0.775V (RMS)

3.2.3 Digital level: dB FS

When audio entered into the digital world other measures were needed. In the early 80's the dB FS (Full Scale) was introduced, in Figure 3-6 we can see a very simplified sampling of analog audio signal using 3 bits.

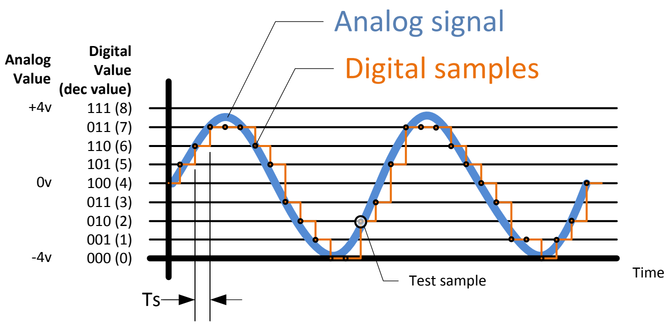


Figure 3-6: Simplified audio sampling using 3 bits

The audio level (A_{level}) in dB FS of a sample is defined as follows:

$$A_{level}\ [dB\ FS] = 20Log_{10}\ \frac{val}{MaxVal} \qquad \text{3-11}$$

Where: **val** = sample value to convert, **MaxVal** = Maximum sample value available in current scale

For instance, to compute the audio level in dB FS of **test sample** in Figure 3-6, applying the equation *3-11* we can obtain that:

$$20Log_{10}\ \frac{Val}{MaxVal}\ = 20Log_{10}\ \frac{2}{8}\ = \mathbf{-12\ dBFS}$$

3.2.4 Audio level references

Years ago too many audio level meters coexist around the world; every meter had its own reference level, units, response time, etc… Due to this many audio level problems existed in international programme exchanges.

In the Figure 3-7 some standardized audio meters with its scales are showed.

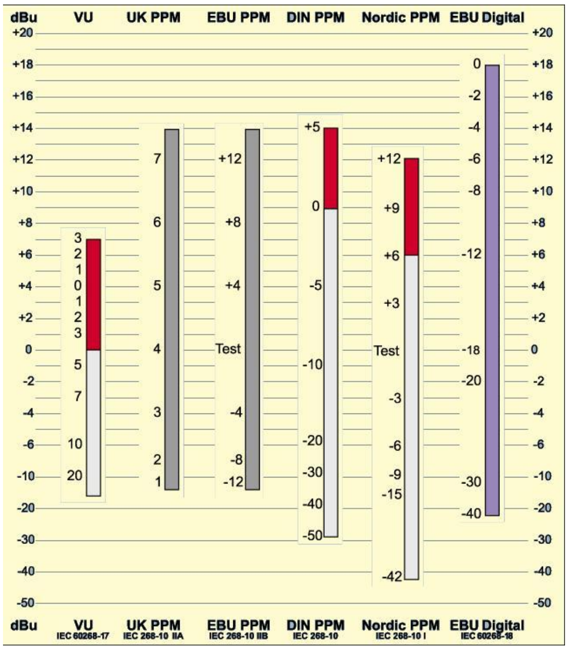


Figure 3-7: Different audio level scales comparison (Picture from [9])

In order to fix this problem ITU published the following recommendation ITU-R BS.645-2 [10], which standardizes the measures, the tools, the units, and the following terms were defined:

- Alignment level (AL): Level of **tone** signal at frequency of 1 kHz which is used to align the international sound-programme connection. The signal level corresponds to 0 dBu.

- Measurement level (ML): Level of a tone signal (12 dB below the AL) which should be **used for long-term measurements** and measurements at all frequencies.

- Permitted maximum level (PML): Level of a tone signal at 1 kHz, 9 dB above the AL. The sound-programe signal should be controlled by the sending broadcaster so that the **amplitudes of the peaks only rarely exceed this level**.

Years later the ITU published ITU-R BS.1726 [11] that is the translation of ITU-R BS.645-2 [10] into digital world:

- Alignment level (AL): -18 dB FS (measured with quasi peak programme meter QPPM)

- Measurement level (ML): -18 dB FS. (measured with QPPM)

- Permitted maximum level (PML): -9 dB FS. (measured with QPPM)

3.2.5 Dynamic range (sound and audio)

The sound dynamic range is the difference between the minimum audible sound to the loudest audible sound without pain, see Figure 3-8.

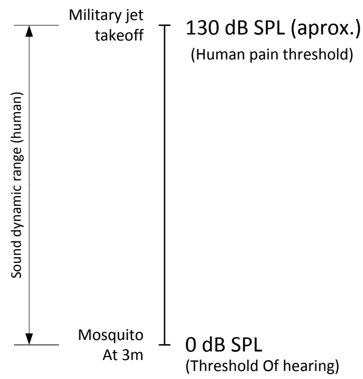


Figure 3-8: Dynamic range

When the sound is recorded, the sound engineers try to use all potential dynamic range that the recording system can offer in order to capture from the quiet to loudest sound, and they use to use some headroom to prevent overloads, see Figure 3-9.

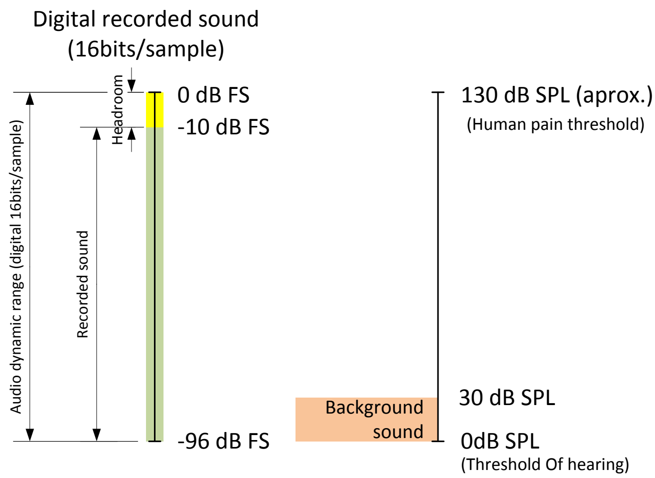


Figure 3-9: Audio dynamic range in recording systems vs sound dynamic range

A recorded sound could be played in many different places with many different audio context conditions. For instance if the sound is played in a cinema, a quiet place with a good calibrated sound system, the useful dynamic range will be large. On the other hand if the sound is played by a car audio system, in general a car is a place where we don't want to listen loud sounds and where the background sound (traffic) is normally loud, the useful dynamic range will be smaller. See Figure 3-10.

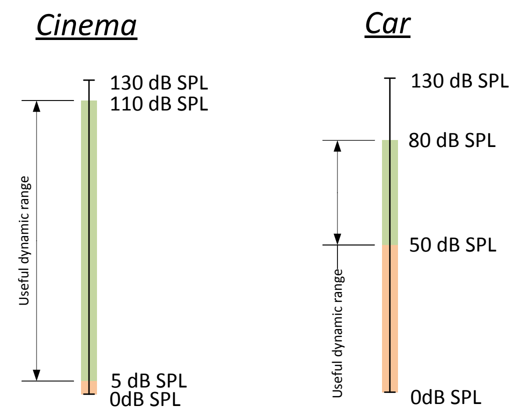


Figure 3-10: Different sound dynamic ranges depending on playing context

Using audio processors or non-linear amplifiers the audio engineers can adapt the recorded sounds to the context which they will be played. For example if a recorded advertisement will be played by a radio station in which the target are drivers, the audio of that advertisement will be compressed (quiet sounds will be amplified and loud sounds will remain equal or will be little attenuated). Doing that, they achieve that quiet dialogs in the advertisement are audible in a car.

4 Audio level meters

In this chapter we describe the most used audio meters. We have divided them into three big groups: short term level meters, long term level meter, and other kind of meters that are out of the scope of this project.

4.1 Short term audio level meters

The audio level meters presented in this section measure the audio level based in a short temporal part of audio signal, typically less than 1 second.

4.1.1 Peak Program meter (or PPM)

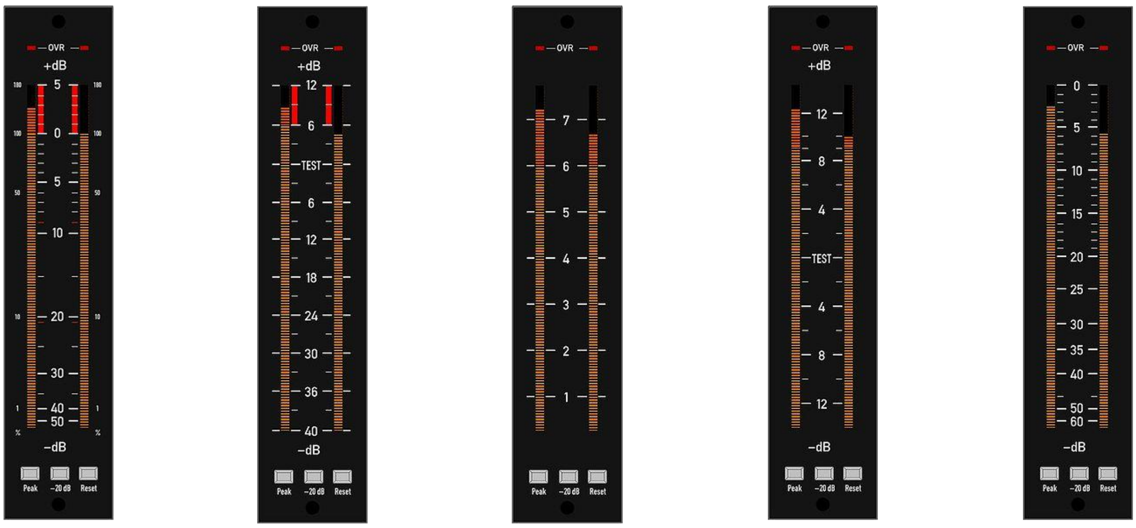
The PPM is an old instrument used in broadcast audio to indicate the instantaneous value of the audio signal, see Figure 4-1. It was designed for analog audio, but it is still used for digital audio signals.

There are 3 important parameters to characterize a PPM:

- Scale: The scale that the user sees and where the signal is measured.

- Integration time: Minimum period during which a sinusoidal voltage should be applied to the instrument for the pointer to reach the 80% of peak value aprox. [10].

- Time to fall: Time for the indicator to fall a determinate number of dB.



PPM IEC 60268-10 I DIN	PPM IEC 60268-10 I Nordic	PPM IEC 60268-10 IIa British	PPM IEC 60268-10 IIb EBU	PPM IEC 60268-18 Digital

Figure 4-1: Different PPM scales (Picture from [12])

There are many standards for PPMs; the differences between standards are scale, integration time, and the time to fall.

For example the PPM IEC 60268-10 II2b (standardized by EBU Tech 3205 [13]):
- Scale: -12dB to +12dB, the center : “TEST” point = 0dBu
- Integration time: 10ms
- Time to fall: 2.8s to fall 28dB

The PPM, in its digital version, is widely used in the digital audio world (IEC 60268-18 [14]).

4.1.2 Volume Unit Meter (VU-Meter)

The VU meter is an instrument which is used to read the mean level of an audio signal. The integration time is much higher than PPM (typically 300ms), and it gives an idea of the mean level of the audio signal, not about the peaks as PPMs.
The first VU-Meter was an electromechanical element (see Figure 4-2).

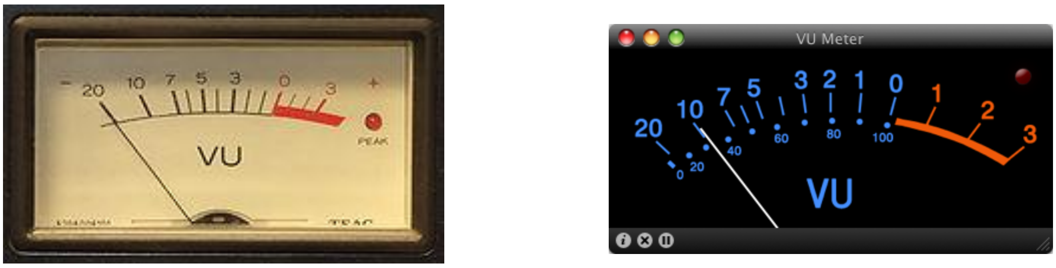


Figure 4-2: **Images of old electromechanical *VU-Meter and a new software VU-meter***

The principal parameters to define a VU-Meter are very similar as the parameters to define a PPM:

- Scale: The scale that the user sees and where the signal is measured.

- Integration time: The time it takes for the needle to reach 99% of the distance to 0 VU when the VU-meter is submitted to a signal that steps from 0 to a level that reads 0 VU.

- Time to fall: Time for the indicator reach the 99% of the distance to 0 VU to a minimum value when the stimulus is switched off.

For example the standard IEC 60268-17 [15] defines:

- Scale: In dB: -20 VU to +3 VU, with 0 point = +4dBu
- Integration time: 300ms
- Time to fall: 300ms

4.1.3 True peak meter

In 2006 the ITU-R BS.1770 [16] standardizes the “True Peak Meter” that is an evolution of old peak meters (see Figure 4-1).
Recently the ITU-R BS.1770-3 [17] have been refined the method to measure the peaks of the true peak (TP) meter.

The old peak meters in the digital world are sample based, they works comparing the absolute value (rectified) of each incoming sample with the meter’s current reading, if the new sample is larger it replaces the current reading, if not, the current reading is multiplied by a constant less than 1 to produce a logarithmic decay.

It is well known that the old peak meters have the following problems:

- Inconsistent peak readings: An analog sequence can produce different peak readings every time that it is played into the digital system.

- Unexpected overloads: Sampled signals may contain overloads even when they have no samples near digital full scale. This overload may appear into subsequent process, for instance: D/A conversion.

- Under reading: This problem occurs when the real peak of the signal is between two digital samples.

In the Figure 4-3 we have illustrated these problems:

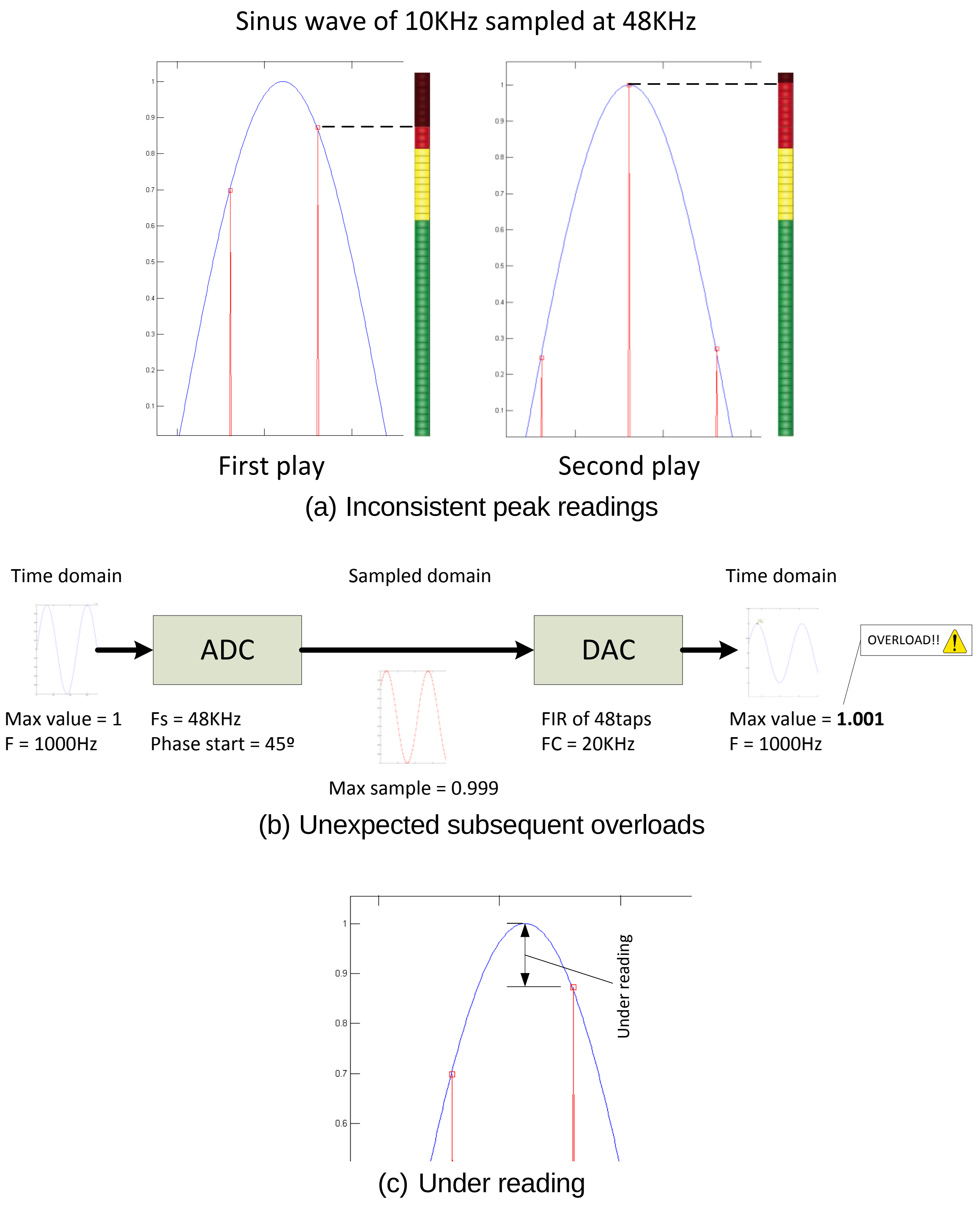


Figure 4-3: Problems of old PPM into digital world

The ITU-R BS1770-3 defines the true peak level as the maximum (positive or negative) value of the signal waveform **in the continuous time domain** and as we have seen, this value may be higher than the largest sample value in the time sampled domain. In order to solve that they propose the following implementation for True Peak meter, see Figure 4-4.

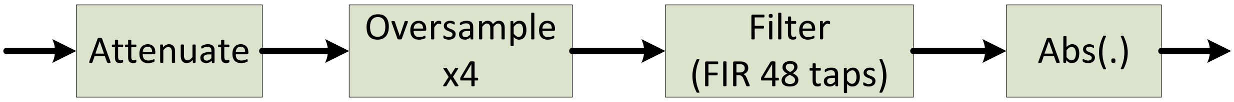


Figure 4-4: True peak meter block diagram (ITU-R BS.1770-3)

In the first stage the signal is attenuated, this step is only necessary to provide enough headroom for the subsequent signal processing. If we use floating point calculus this step is not necessary.
In the second stage the signal is oversampled by factor of 4; this is done by adding three cero samples after each real sample.
After that, a low- pass filter is applied to the signal in order to obtain an accurate approximation of time domain waveform.
In the last stage the absolute value of signal is obtained.

In the Figure 4-5 a commercial True Peak meter is shown.

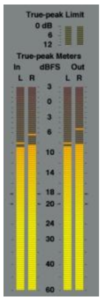


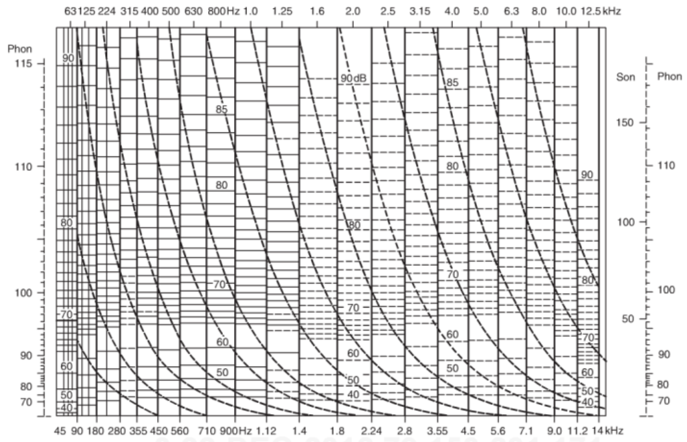
Figure 4-5: Real True peak meter, TCelectronic LM2 (Picture from [18])

4.2 Long term audio level meters (Loudness meters)

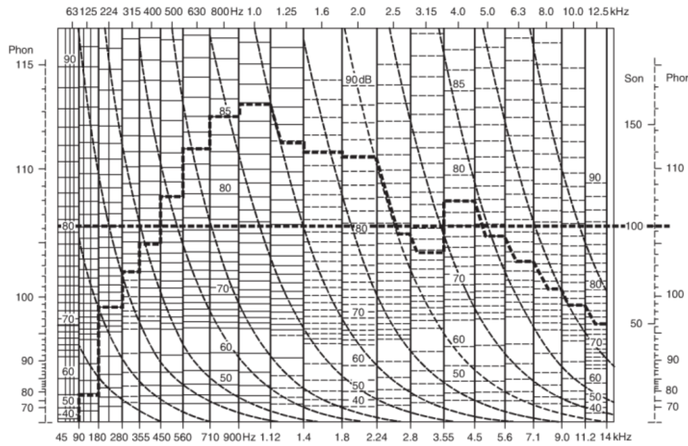
In order to obtain a measure that can indicate the loudness of a complete content (programme, advertisement, promo, etc…) a long term measure is needed. The key idea is to obtain a method to calculate a value proportional to the loudness of the analyzed content. Then, comparing those values we can classify the contents from loudest to most quiet, and we can compare the loudness of that contents to a reference level knowing how far are from that reference.

4.2.1 Zwicker loudness method

Around 1960 E. Zwicker published a method to determine the loudness of a programme, in 1975 that method was published as the standard ISO 532-1975 [19].
The Zwicker method was based on filling a chart, different charts exists depending on sound nature.
The horizontal axis of the charts is the audio spectrum divided into boxes, and in vertical axis there are the sound level in phons (Figure 4-6 a).
The highest values into each spectrum division must be plotted in the chart. Then the points have to be connected respecting the decay lines. Finally a horizontal line have to be drawn where the area that is inside the curve above the line is of the same magnitude as the area that is outside the curve under the line. The point where intersects the scale is the loudness measure (Figure 4-6 b)



(a) Empty chart



(b) Filled chart

Figure 4-6: E. Zwicker loudness C8 chart (Picture from [20])

Several revisions and simplifications of this method have been published since it has been released.

4.2.2 Equivalent level (Leq)

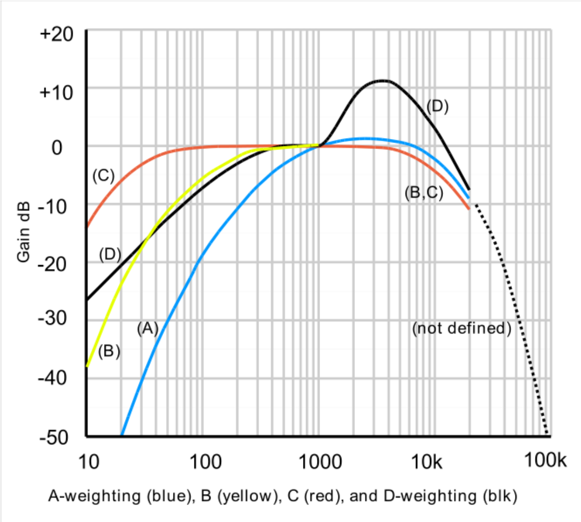
This method defines the equation *4-1* (energy based) to obtain a single value equivalent to the loudness of whole audio sequence.

$$\mathrm{Leq}\left(\underline{w}\right)\ [\mathrm{dB}\underline{w}] = 10Log_{10}\left(\frac{1}{T}\int_0^T \frac{Xw^2}{Xref^2}dt\right) \qquad \textit{4-1}$$

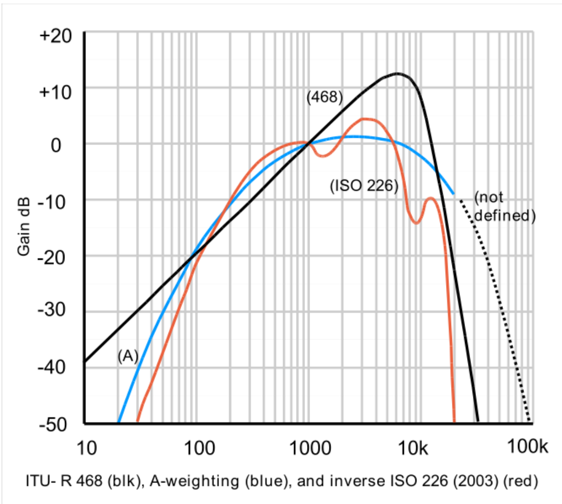
Where: **w**= Frequency weighting method, **T**= length of audio sequence, **Xw**= Signal at the output of weighting filter, **Xref**= Some reference level

Usually a frequency weighting (w) is used before calculate the Leq. In the Figure 4-7 we can see the most used frequency weighting curves: IEC-A/B/C/D [21], CCIR 468 [22], ISO 226:2003 [7], and RLB [17].

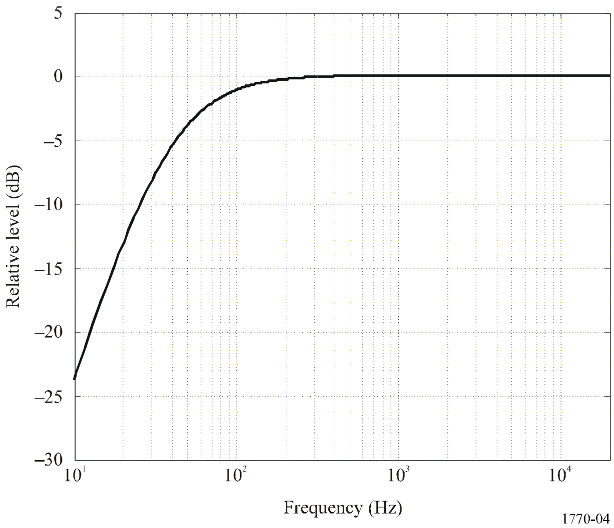
The correct notation for the Leq using IEC-A is **Leq(A).**



(a) IEC curves (Picture from [23])



(b) CCIR 468 and ISO 226 (Picture from [24])



(c) RLB curve (Picture from [17])

Figure 4-7: Frequency weighting curves

Dolby laboratories promoted a measure named Leq(m) that was designed to measure the Loudness in cinemas (films, trailers, etc…). It uses CCIR 468 frequency weighting and all audio channels are taken into account.

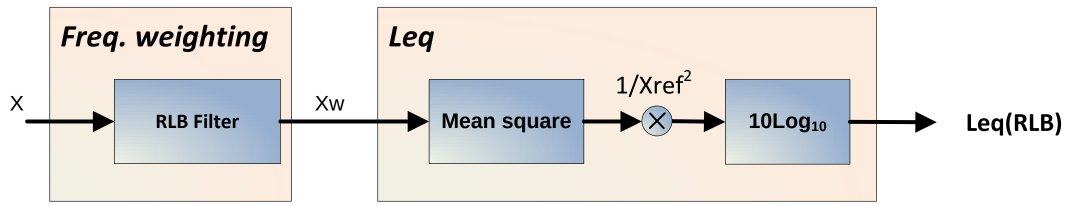
4.2.3 ITU-R BS.1770

In order to develop a world standardized loudness meter for the media industry, the ITU used the paper “Evaluation of Objective Loudness Meters” written by Gilbert A. Soulodre in 2004 [25]. In this paper 10 loudness meters were evaluated using a real human audience.
The method with the best performance was the Leq(RLB) and that method was the chosen one to use as a base for the ITU-R 1770 [16].
The objectives that ITU wanted to reach with their recommendation were: accuracy, multichannel, and simplicity (low implementing cost). With these ideas in mind in 2006 the ITU-R 1770 was published.
To add the multichannel feature to Leq(RLB) meter, they had to make some changes in the original meter design:

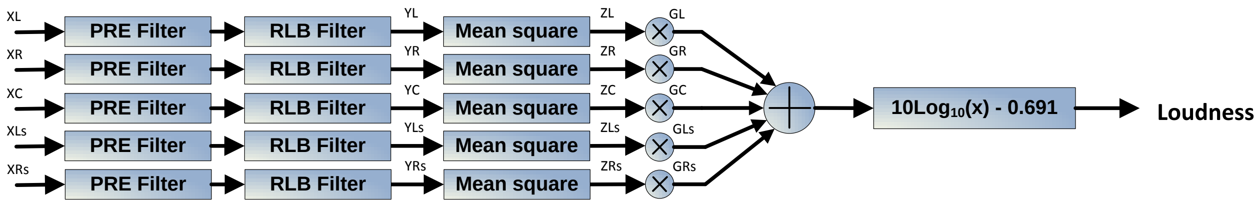
- Add a first stage shelving filter to model the acoustic effect of the head.

- Add a final weighted sum in order to obtain a single loudness value in multichannel environments.

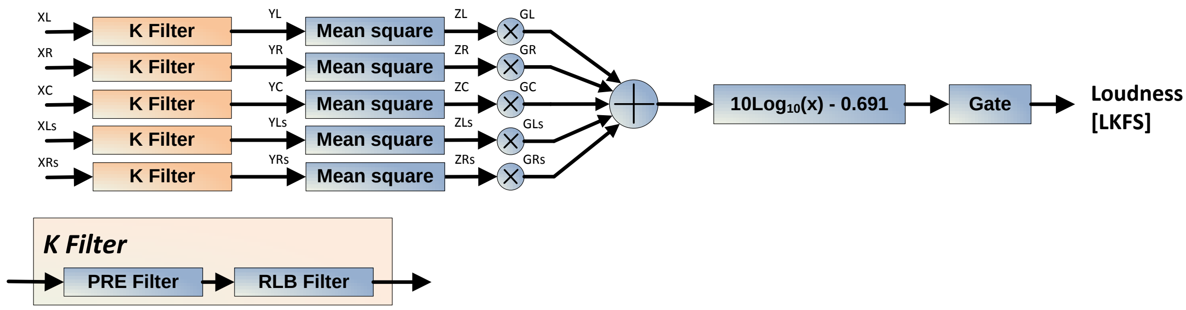
The Figure 4-8 shows a basic Leq(RLB) implementation and different ITU-R BS.1770 implementations.



(a) Leq(RLB)



(b) ITU-R BS.1770 [16]



(c) ITU-R BS.1770-3 [17]

Figure 4-8: Leq(RLB) and ITU-R BS.1770 implementations

The “mean square” block implements the following equation:

$$\mathrm{MS}(\mathrm{x}) = \frac{1}{T}\int_{0}^{T} x^2 dt \qquad \textit{4-2}$$

In the following revisions some changes were introduced to ITU-R BS.1770, the most significant ones will be announced below:

- ITU-R BS.1770-1 [26]:
    - The logarithmical unit LKFS was announced as loudness measure instead of dB.

- ITU-R BS.1770-2 [27]:
    - It applies a fixed gate in order to avoid the silent parts of the content in the loudness computation.
    - It uses a relative gating algorithm to eliminate the relative quiet parts of the content of the loudness calculation.
    - The implementation of gating algorithms are done comparing loudness of audio sample blocks of 400ms overlapped 75%.

- ITU-R BS.1770-3 [17]:
    - This revision only affects the true peak meter definition, see 4.1.3.

In chapter 7 several implementation details of ITU-R BS.1770-3 [17] will be showed.

4.2.4 Dobly AC-3 and Dialnorm

The digital audio compression standard (AC-3), so called Dolby AC-3 [28] is not a loudness meter itself, but it has the dialnorm parameter that allows it to control the playback loudness of the AC-3 encoded audio streams.
The Dolby AC-3 is a digital audio compression system that accepts to compress 5.1 audio streams (6 audio channels) into a 384Kbps data stream. It is an ideal system to use in satellite links or digital terrestrial televisions (DTT) stations where the bandwidth is a big concern.
A goal of this system is to transmit the audio signals as close to original as possible, and it adds metadata to complement the compressed audio signal. This extra metadata provides the system of following features:

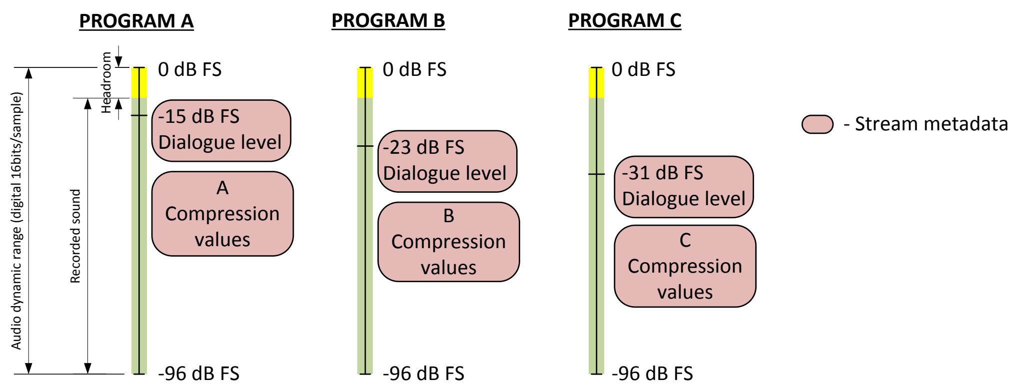
- Downmixing: Allows a smart conversion of original audio stream to less channels than original.

- Dinamic Range Compression (DRC): Adapts the audio to the listener system conditions.

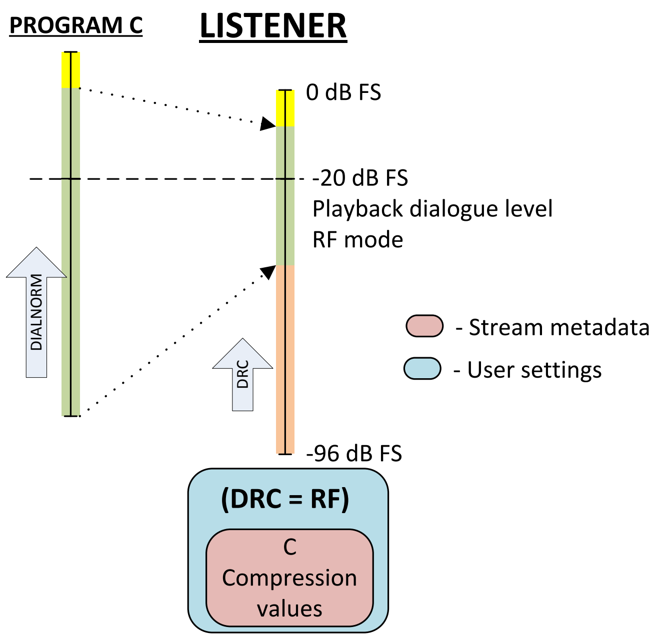
- Dialog normalization (dialnorm): It changes audio playback level of the listener system according with the metadata inside the delivered audio stream. If this metadata has been computed properly the listener system will equal the loudness level of all channels and contents.

The dialnorm feature deserves an extended explanation due to it plays an important role in loudness standardization. Dialnorm means dialog normalization and it is a different approach to loudness normalization.
The key idea about the use of dialnorm parameter is to preserve the original dynamic range (see 3.2.5) by adding a metadata value into digital audio stream that indicates at which level is recorded a normal dialog. The receiver reads that metadata (integer value between -1 and -31) and according to its configuration parameters, the receiver shifts the audio level preserving the dialog level between programs [29].

The Dolby digital compliant receivers also have a DRC system that allows broadcasters to send a full dynamics version of its contents. This system does the level shifting that is read from dialnorm value and it corrects as well the playback dynamic range according to the audio stream metadata and to the DRC user settings (Line or RF mode). See Figure 4-9.



(a) Recoded program levels



(a) Playback program level shifted and compressed

Figure 4-9: Dialnorm

4.3 Other audio meters

Other kinds of audio meters exist but they are not directly involved in the loudness normalization process. Some of them are announced in following lines.

4.3.1 Spectrum analyzers

This kind of meters analyzes the frequency components of an audio signal. See Figure 4-10.

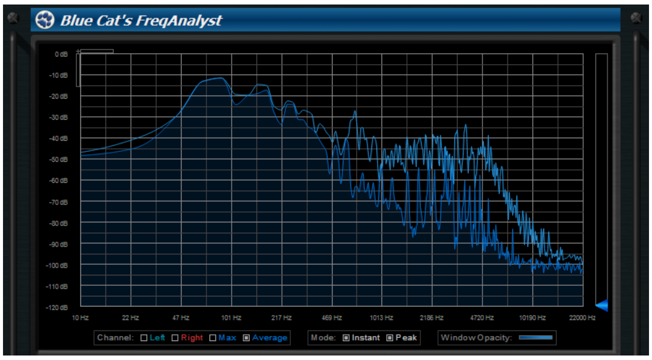
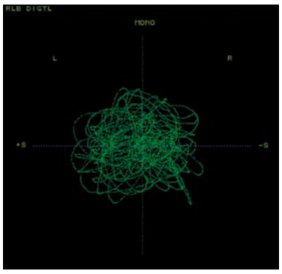


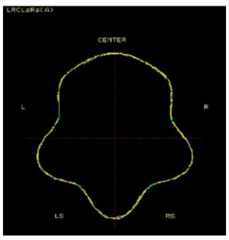
Figure 4-10: Audio spectrum analyzer (Picture from [30])

4.3.2 Phase meters

When more than one audio channel has to be used, the audio align problem appears, and the sound engineers needs tools to measure the phase between audio signals, some of these tools are showed in Figure 4-11.



(a) Audio vectoroscope (Picture from [31])



(b) Jellyfish display (Picture from [32])

Figure 4-11: Audio level align and phase meters

5 Loudness standards

Nowadays two popular loudness standards coexist in the word, the EBU R 128 [2], and A/85 [33], both are similar and they are based on ITU-R BS.1770 [16].
The R 128 was published by EBU in 2011 and it is being adopted by countries from EU. And A/85 was published in 2009 by ATSC and is the technical leg of CALM law [1]. The most important countries that use A/85 are USA and Canada.
Both standards use the loudness meter and the true peak (TP) meter defined in ITU-R BS.1770 [16]**.**

The main differences are:

- The EBU R 128 sets the loudness target level to -23.0 LUFS(*) and A/85 sets the loudness target for delivery and content exchange to -24 LKFS(*).

- The EBU R 128 uses a loudness gated measure (ITU-R BS.1770-2 [27]), and the A/85 does not (ITU-R BS.1770-1 [26]).

- The A/85 promotes the Dolby AC-3 [28] proprietary system as standard DTT audio encoding method. This system implements another loudness control approach: Send the original untouched audio data, add the necessary metadata to the stream, and use that metadata and listener receiver configuration to process the played audio.

- The A/85 focuses to the anchor element to measure the loudness of a program, usually the anchor element are dialogs. In the EBU R 128 the anchor element does not exist, instead it uses a relative threshold to avoid quiet parts of loudness in the mean measure calculus.

(*) LUFS is the same as LKFS, only changes the name.

5.1 EBU Recommendation 128

The R 128 [2] was published by EBU in 2011 as a recommendation, and its title was "Loudness normalization and permitted maximum level of audio signals". It is based on the loudness meter defined by ITU-R BS.1770-2 [27].
The R 128 contains references to different EBU technical documents in order to complement it; the most important are listed below:

- **EBU Tech 3341** [34]**:** "Loudness Metering: 'EBU Mode' metering to supplement loudness normalization in accordance with EBU R 128"
  Define with detail the characteristics that must comply a loudness and TP meters to work in "EBU Mode": Algorithms, time scales, units, display, and calibration.
  This document is based on ITU-R BS.1770-2 loudness and TP meter definitions.
  The same document provides us with test material in order to check the implementation accuracy of our loudness meter.

- **EBU Tech 3342** [35]**:** "Loudness Range: A measure to supplement loudness normalization in accordance with EBU R 128"
  This document defines the loudness range (LRA) measure and explains how to implement a LRA meter properly.
  The LRA quantifies the variation of loudness measure in the specified portion of time.
  This document also provides us with test material in order to check the implementation accuracy of our LRA meter.

- **EBU Tech 3343** [36]**:** “Practical guidelines for Production and Implementation in accordance with EBU R 128”
  This is a document addressed to broadcasters or companies that produce professional audiovisual media. It explains the loudness motivation and provides them with a good practice guide.

- **EBU Tech 3344** [37]**: “**Practical guidelines for distribution systems in accordance with EBU R 128”
  It explains how to design broadcast distribution systems if we want to properly accomplish the R 128.

In order to summarize the R 128, the most important parts are listed below:

Loudness and TP measuring algorithms:

- ITU-R BS.1770-2 [27] (with absolute and relative gating).

Definitions:

- Units: The LKFS loudness units defined in ITU-R BS.1770-1 [26] change its name to LUFS. The LU is defined as a relative LUFS unit.

- Time scales: Three different loudness measuring time scales are defined:
  - **Momentary**: It uses a sliding window of 400ms with 75% of overlapping factor. The measurement is not gated.
  - **Short-term**: It uses a sliding window of 3s and the measurement is not gated, and no overlapping factor is defined. But if you want to use this measure as base to compute the LRA in EBU Tech 3342 a minimum overlapping factor of 66% is announced.
  - **Integrated**: It is relative to all programme duration and the measure is gated according to ITU-R BS.1770-2.

- Programme Loudness: The integrated loudness over the duration of a programme (in LUFS).

- Loudness Range (LRA): It describes the distribution of loudness within a programme or portion of time.

- Maximum True Peak level: The maximum value of the audio signal waveform of a programme in the continuous time domain.

Main recommendations of R 128 are:

- The measurements shall be made with a loudness meter compliant with EBU Tech 3341 (Loudness and TP), and EBU Tech 3342 (LRA).

- The measures programme Loudness, Loudness Range and Maximum True Peak Level shall be used to characterize an audio signal.

- The programme Loudness Level shall be **normalized to a target level of -23.0 LUFS**. The permitted deviation from the target level shall generally not exceed ±1.0 LU for programmes where an exact normalization to target level is not achievable practically (for example, live programmes).

- The maximum **permitted TP level of a programme during production shall be -1 dBTP.**

- The audio signal shall generally be measured in its entirety, without emphasis on specific elements such as voice, music or sound effects.

This recommendation has been highly accepted by broadcast community of around the world; it has become a standard in EU and most of its countries are adopting it in a different ways, for instance: France as a law, Germany as a directive, Spain as a recommendation, etc…

Nowadays, the majority of broadcast technology vendors integrate the R 128 measures into their devices: Tecktronic, Nungen audio, DK-Technologies, Harrys, Yamaha, Orban, Lawo, etc…

5.2 Recommended practice A/85

The recommended practice (RP) A/85 [33] was published in 2009 by Advanced Television Systems Committee (ATSC) under CALM law [1] umbrella. In July of 2011 the amendment 2 of RP A/85 was approved. Its complete name is “Techniques for Establishing and Maintaining Audio Loudness for Digital Television”.
This RP A/85 is focused on the DTT system and provides definitions, rules, and recommendations to produce, distribute, and delivery television content.
In USA the DTT system was standardized by A/53 part 1 [38] “Digital television standard”; in this paper it is said that the audio streams have to be encoded using the Dolby AC-3 [28] system (see the standards A/53 part 5 [39], and A/53 part 6 [40]).
As it is explained in section 4.2.4, the Dolby AC-3 system integrates a different approach to equal loudness level between channels and programmes. Its idea is to transmit the original audio signals with metadata and apply a level shift (based on metadata and receiver configuration) on viewer side in order to match audio levels.

In order to summarize the RP A/85, the most important parts are announced below:

Loudness and TP measuring algorithms:

- ITU-R BS.1770-1 [26] (without gating).

Definitions:

- Units: The LKFS loudness units are defined in ITU-R BS.1770-1.

- Anchor element: The perceptual loudness reference point, usually the dialogs.

Underlying audio codec:

- Dolby AC-3 (see 4.2.4).

The RP A/85 main recommendations are:

- That the measurements shall be made with a loudness meter compliant with ITU-R BS.1770-1 (Loudness and TP).

- That for delivery or exchange of content **without metadata** the target loudness should be **-24 LKFS ±2dB**.

- That for delivery or exchange of content **with metadata** the dialnorm parameter has to be set properly.

- That the maximum **permitted TP level of a programme shall be -2 dB TP.**

The RP A/85 also includes information about the following subjects:

- Set up reference monitoring environments.

- Methods to effectively control program-to-interstitial loudness.

- Effective uses of audio metadata for production, distribution, and transmission of digital content.

- Affectation of DRC in loudness.

The A/85 is used mainly in USA and Canada.

5.3 ARIB TR-B32

In March of 2011 the ARIB (Association of Radio Industries and business) in Japan published the document TR-B32: “Operational Guidelines for loudness in broadcast digital television”. This document compiles state of the art of loudness technology around the world. And it could be used as loudness reference document for Japanese broadcast vendors.

6 State of the art of loudness meters

In this section we analyze the state of the art of loudness meters; we can see and compare the implemented standards, the allowed input signals, the prices, etc…
We will divide them into three big groups: standalone loudness meters, software based meters, and the last group will be filled by audio mixers with integrated loudness meters.

Nowadays loudness is becoming a common subject when we talk about professional audio, and many manufacturers are producing equipment related to it. We summarized trying to select the most remarkable loudness meters in broadcast market.

If you want further information about companies that produce loudness meters, the EBU provides an updated web page with a list of EBU R 128 implementers [41].

6.1 Standalone loudness meters

This kind of loudness meters are usually used in master control rooms, quality control sites, or sound mixing cabins where the mixer does not have integrated a loudness meter.

They can be characterized by: The types of input signals, the parameters that can measure, the number of audio channels that can analyze, and obviously the price.

	*Manufacturer:*	**DK-Technologies**
	Model:	**DK1**
	Input signals:	**AES, analog**
	Number channels:	**4**
	Audio Measures:	**Loudness (LRA, Shortterm, Momentary), True Peak, PPM, Vectorscope, Phase correlation**
	Loudness standards:	**BS.1770, A/85, R128**
	Price:	**1.000€**
	Description:	Portable digital/analog loudness meter

	*Manufacturer:*	**Tecktronix**
	Model:	**WVR8300**
	Input signals:	**CBVS, SDI, AES, Analog,**
	Number channels:	**16**
	Audio Measures:	**Loudness (LRA, Shortterm, Momentary, Session), True peak, PPM, Vectoroscope, Phase correlation, surround analyzer**
	Loudness standards:	**BS.1770-2**
	Price:	**16.000€ (depending on options)**
	Description:	Complete broadcast video analyzer with audio option

	*Manufacturer:*	**TC Electronic**
	Model:	**LM2**
	Input signals:	**AES, analog**
	Number channels:	**2**
	Audio Measures:	**Loudness (LRA, Shortterm, Momentary, Session), True peak, PPM**
	Loudness standards:	**BS.1770, BS.1770-2, A/85, R128**
	Price:	**2.200€**
	Description:	Simple audio meter and standard transcoder

	*Manufacturer:*	**Harris**
	Model:	**CMN-LA**
	Input signals:	**SDI, AES, Analog**
	Number channels:	**16**
	Audio Measures:	**Loudness (LRA, Shortterm, Momentary, Session), True peak PPM**
	Loudness standards:	**BS.1770, BS.1770-2, A/85, R128**
	Price:	**5.000€**
	Description:	Complete portable loudness meter

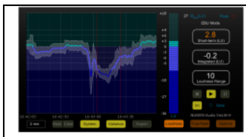
	*Manufacturer:*	**Dolby**
	Model:	**LM100**
	Input signals:	**AES, Analog**
	Number channels:	**6**
	Audio Measures:	**Loudness (LRA, Shortterm, Momentary, Session), True peak, PPM**
	Loudness standards:	**BS.1770-1, BS.1770-2, Leq(A), R128**
	Price:	**2.600€**
	Description:	Loudness meter

6.2 Software loudness meters

This kind of loudness meters have to run on a computer and its audio source are the media files in that computer or any audio capture device installed in it (DTT capture card, audio IO board, SDI input board, ethersound protocol, etc…).

We can characterize the software loudness meters by the type of solution (complete app, library, or filter-plug-in), the measures that it can do, and the price.

	*Manufacturer:*	**DVBControl**
	Model:	**DVBLoudness**
	Platforms:	**Windows**
	Solution type:	**Complete app**
	Audio Measures:	**Loudness (LRA, Shortterm, Momentary), True Peak, PPM**
	Loudness standards:	**BS.1770-3, R128**
	Price:	**4.500€ (to measure 25 audio channels)**
	Description:	Application that analyzes with detail the broadcasted signals. Allows reporting. Part a complete broadcast analysis suite

	*Manufacturer:*	**NUGEN Audio**
	Model:	**VisLM-H**
	Platforms:	**Windows, MAC, VST, AU**
	Solution type:	**Complete app and plug-in**
	Audio Measures:	**Loudness (LRA, Shortterm, Momentary), True Peak PPM**
	Loudness standards:	**A/85, R128**
	Price:	**350€**
	Description:	Small application / plug-in that analyzes an input audio signal or a file

	*Manufacturer:*	**SourceForge (GNU)**
	Model:	**R128GAIN**
	Platforms:	**Independent**
	Solution type:	**Library**
	Audio Measures:	**Loudness (LRA, Shortterm, Momentary), True Peak, PPM**
	Loudness standards:	**A/85, R128**
	Price:	**0€**
	Description:	Free library that can be used to calc loudness of a file

	*Manufacturer:*	**Hamlet**
	Model:	**Guardian**
	Platforms:	**Windows**
	Solution type:	**Complete app**
	Audio Measures:	**Loudness (LRA, Shortterm, Momentary), True Peak, PPM**
	Loudness standards:	**R128**
	Price:	**315€**
	Description:	Small application that analyzes an input audio signal or a file

6.3 Audio mixers with loudness meters

Nowadays only lawo build an audio mixer with loudness meter integrated in it. It is a very helpful feature for sound engineers because they do not need ancillary equipment to measure the audio loudness.

	*Manufacturer:*	**Lawo**
	Model:	**MC^{2}90, MC^{2}66, MC^{2}56**
	Audio loudness measures:	**Loudness (Shortterm, Momentary), True Peak, PPM**
	Loudness standards:	**A/85, R128**
	Price:	**Starting at: 18.000€** (depending on configuration)
	Description:	Professional mixers for live production or studio

SECTION 2: Loudness meter implementation

7 Our loudness meter implementation

Starting from ITU-R BS.1770-2 we have developed **from 0** a complete loudness meter and true peak (TP) meter. We have decided to use the ANSI C++ as a programming language (no extra libraries used) in order to allow the compilation of the code in any platform (Linux, MAC OS, Windows, etc…).
To check our loudness/TP meter we have implemented (without libraries) a wav reader, wav writer, and a windows command line wrapper.

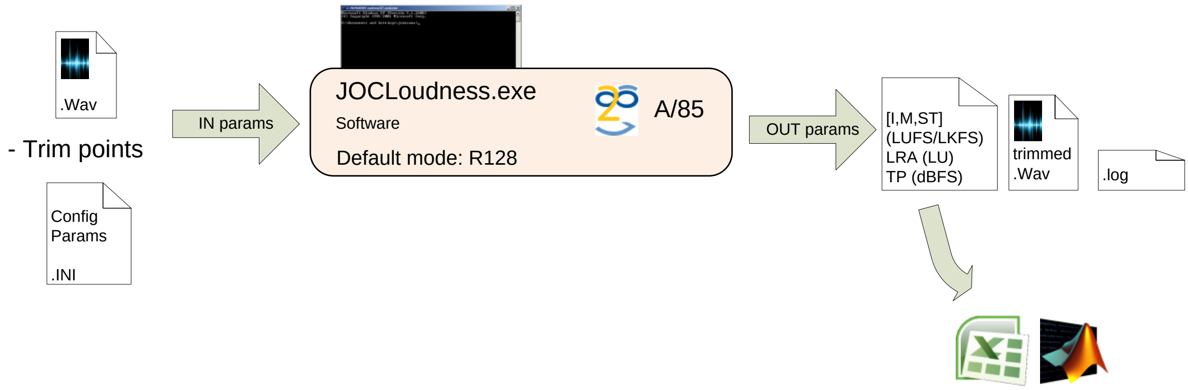


Figure 7-1: Command line loudness/TP meter

As you can see in the Figure 7-1 the command line loudness meter needs an input sound file to analyze (wav format) and it computes the following:

- Integrated loudness value.
- Vector of momentary loudness.
- Vector of short term loudness.
- Vector true peak level.
- Maximum true peak level per channel.
- Loudness range value.

For instance if you analyze the file “seq-3341-7_seq-3342-5-24bit.wav” (extracted from EBU website) using the R128 preset you will obtain the following result:

```
# Loudness data

Integrate value (LUFS) = -23.0

Loudness range (LU) = 4.9

LRA loudness sample interval (secs) = 0.750000
LRA loudness sample interval (audio samples) = 36000
LRA loudness (LU) = [-27.6 -25.8 -24.9 -24.3 … … -21.8 -21.8 -22.6 ]

Momentary loudness sample interval (secs) = 0.100000
Momentary loudness sample interval (audio samples) = 4800
```

```
Momentary loudness (LUFS) = [-27.9 -24.3 -22.5 -21.3 -21.2 … … -25.3 -25.1 -24.9 ]

Short term loudness sample interval (secs) = 0.750000
Short term loudness sample interval (audio samples) = 36000
Short term loudness (LUFS) = [-27.6 -25.8 -24.9 -24.3 -25.9 … … -21.8 -21.8 -22.6 ]

# True Peak values

Max TP value (dBFS) = [-9.1 -8.9 ]

TP sample interval (secs) = 0.250000
TP sample interval (audio samples) = 12000

TP values channel 0 (dBFS) = [-16.3 -16.3 -18.9 -18.4 … … -18.3 -19.8 -17.5 ]
TP values channel 1 (dBFS) = [-16.2 -16.3 -18.4 -18.3 … … -18.3 -20.0 -17.3 ]

(… …) Abbreviated data
```

Figure 7-2: Resulting loudness/TP data

This resulting data can be easily imported to any analysis software (like MS Excel or Matlab) in order to get conclusions, or it can be inserted into a database to store it and compare it with others.

For a complete manual of the command line loudness/TP meter you can see the annex 15.1.

7.1 Loudness Measuring Engine (LME)

The kernel of our loudness monitoring system is this module, the loudness/TP measuring engine (LME). It has been written in ASNI C++ and it does not contain any additional libraries, this allows to easily compile this engine in any platform (Linux, UNIX, Windows, MAC OS, etc…).

The LME can be highly parameterized using the .ini file, but it is important to say that it has 2 internal configured presets in order to avoid the complex parameterization stage, those presets are: R128, and A/85.
If you want more information about the LME parameterization see the annex 15.1.

To skip headroom calculus problems in the LME all input data are transformed into float double precision (64 bits) and all internal calculations are done in that precision.

In the following figures we can see a block diagram of the LME.

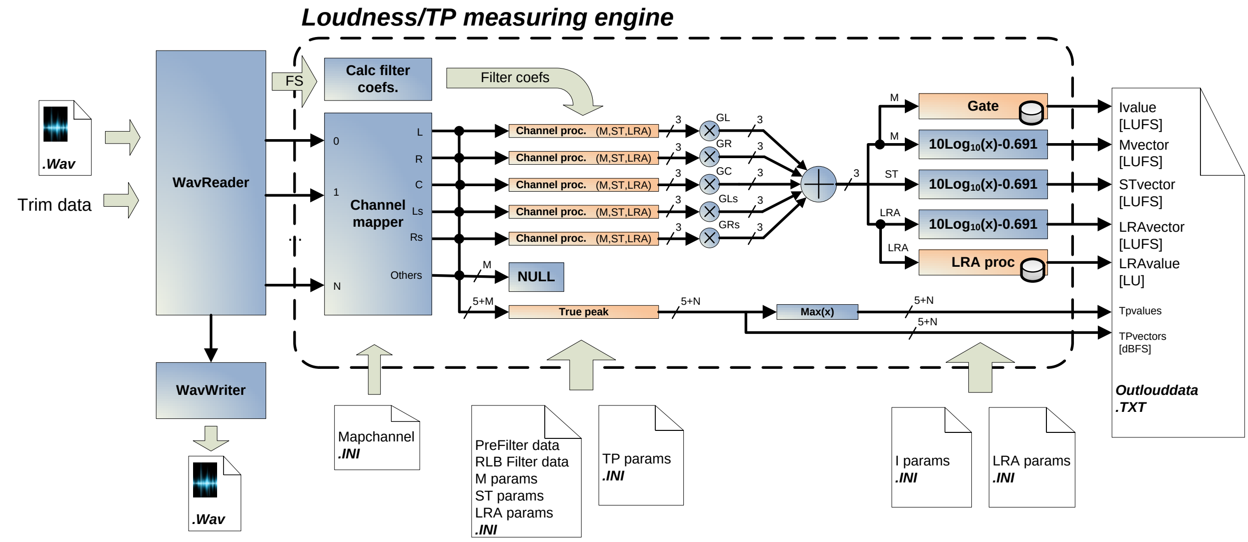


Figure 7-3: Block diagram of the loudness/TP measuring engine (LME)

The first step is to set up all coefficients of implemented filters in LME according to sampling frequency of the audio source. We have to recalculate the coefficients of:

- Pre filter (according to ITU R-BS.1770-2 filter curve).
- RLB filter (according to ITU R-BS.1770-2 filter curve).
- Low pass filter of TP section (according to ITU R-BS.1770-2 definition).

The second step is to set up the *channel mapper* according to configuration file. This module reads the number of audio channels available from source and it maps the input channels into the properly outputs.

Once filters and *channel mapper* are configured, we can start to send audio samples to *channel proc* and *True peak* modules. The samples of channels that could not be mapped are discarded (send to *NULL*).

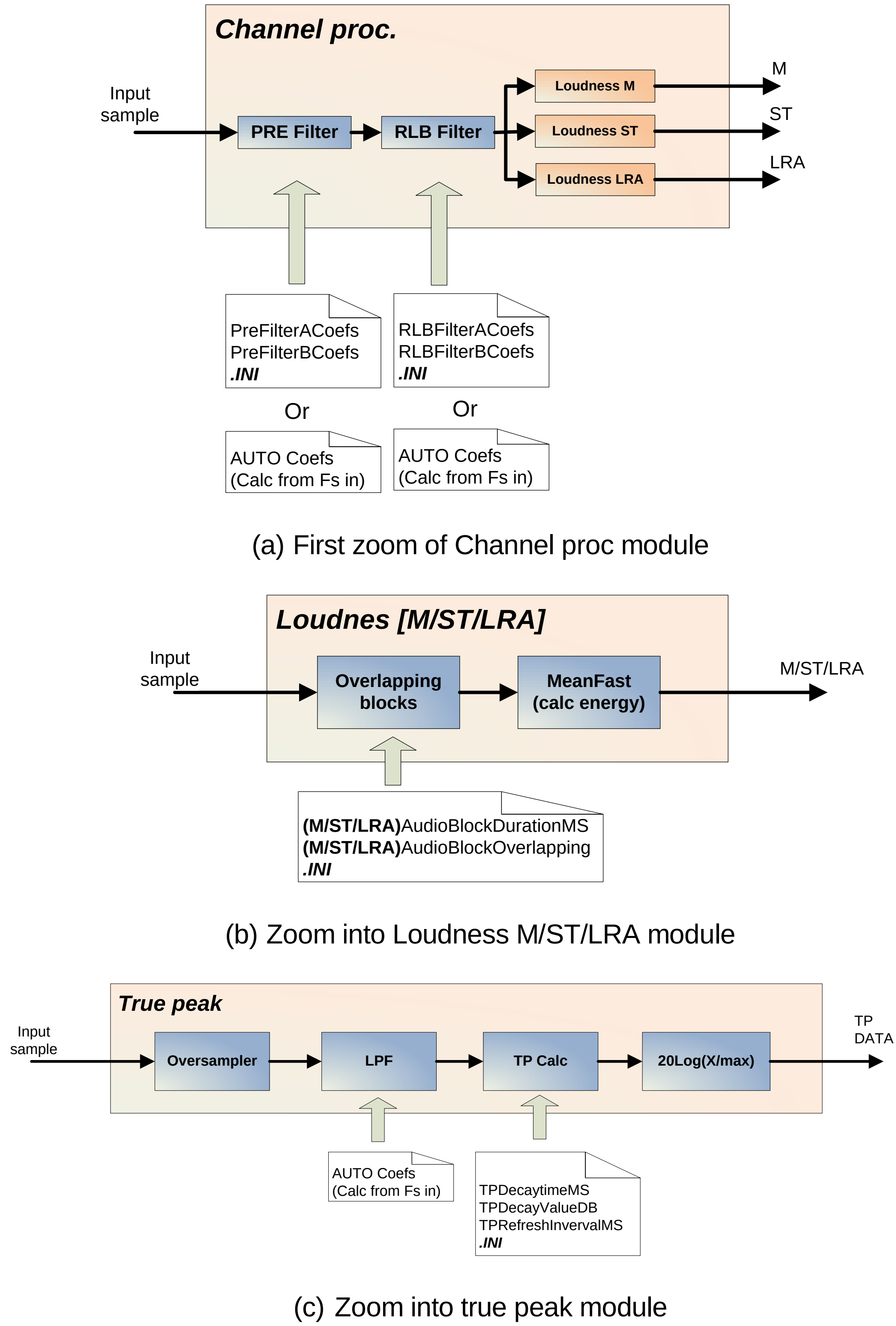


Figure 7-4: Block diagram of the channel proc module

As we can see in Figure 7-4(a), every sample that is sent to *channel proc* module is filtered twice, in the first stage by PRE filter and in the second stage by RLB Filter. After that the resulting samples are sent to 3 different modules: *Loudness M*, *loudness ST*, and *loudness LRA*.

The function of *loudness M/ST/LRA* modules is to compute the energy of the audio signal according to the time and overlapping parameters. The three modules do exactly the same function but with different timing and overlapping values (see Figure 7-4(b)).

In order to compute the true peak data all samples are sent to *True peak* module, see Figure 7-4(c). The first action that is done in this module is to oversample the input data putting 0 after each sample, the oversampling factor is x4 or x2 depending on original sampling frequency.
After oversample stage the resulting signals are filtered by a low pass filter (*LPF*) in order to get a continuous digital signal. Then the samples are computed by *TP Calc* module that extracts the maximum value of the rectified signal, and this value is updated or not depending on the decay values that are set in the configuration.
Finally this value is converted into a logarithmic scale (dBFS).

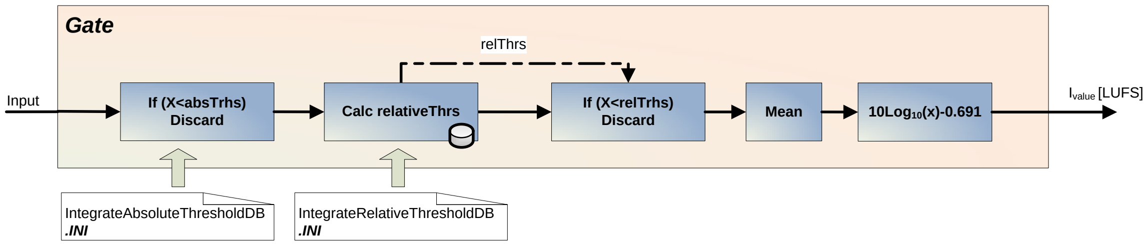


Figure 7-5: Block diagram of the gate module

The gate is defined into ITU-R BS.1770-2 [27], the main idea of this module is to ignore the quiet parts of audio from loudness computation and improve the precision of loudness measure in programmes with a large loudness range.

In the first stage the loudness data that are below certain threshold are discarded. After that the block *Calc realtiveThrs* computes another threshold using all loudness samples of the programme, this threshold is named relative threshold (relThrs). Then all loudness samples of the programme below this relThrs are discarded as well. Finally the mean of all remaining loudness samples are computed and converted into a logarithmic scale, and this value is the integrated loudness or programme loudness.

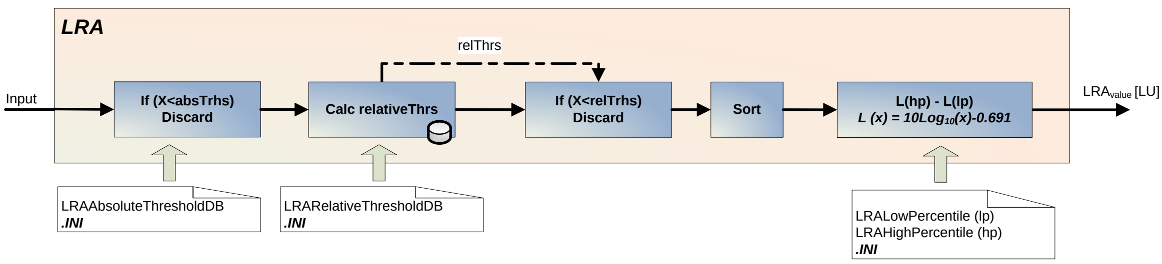


Figure 7-6: Block diagram of the LRA computation module

As we can see in the Figure 7-6 the first three blocks of LRA module are identical to the first three blocks of the Gate module, but the differences start when we have the loudness samples vector, then we have to sort the samples of the vector from highest to lowest, and finally compute the difference between the high and low percentile of the distribution into logarithmical scale, and this value will be the loudness range (LRA).

The operational functions of the LME could be summarized into the following workflow diagram (see Figure 7-7).

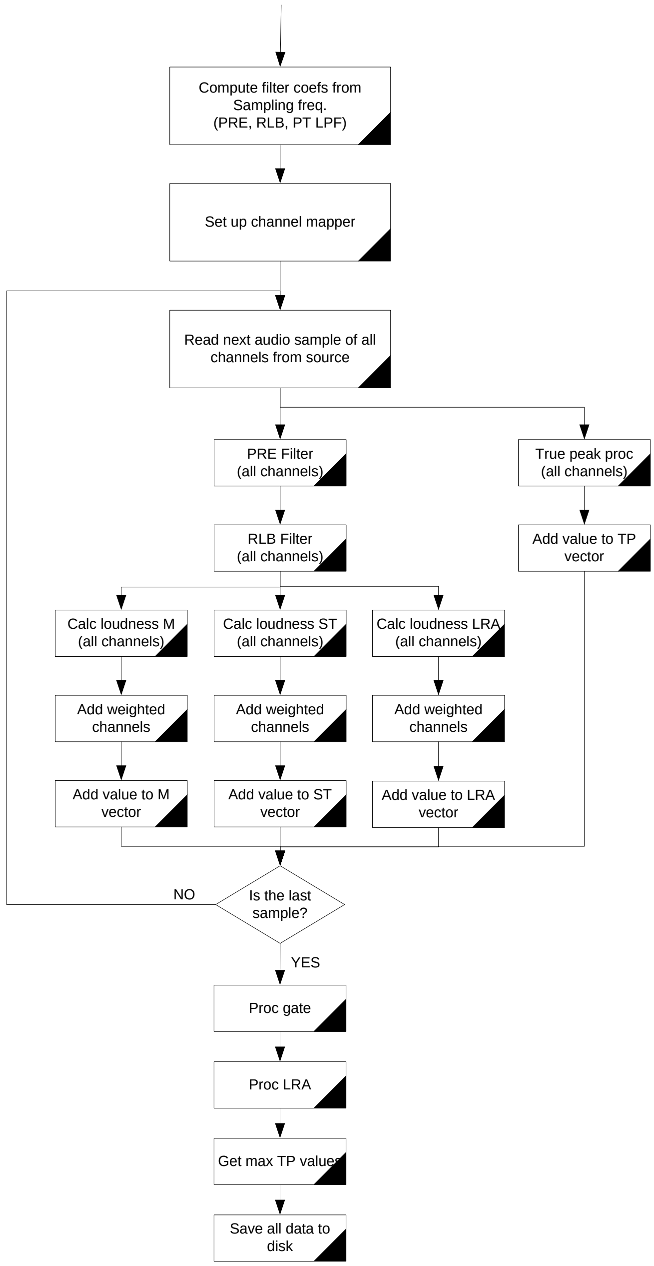


Figure 7-7: Workflows of loudness/TP measuring engine

It is important to say that the architecture of the measuring engine allows it to run in real time mode (samples are sent while they are generated) or in offline mode (all samples are sent at the same time).

7.1.1 Performance

In order to check the performance of the LME we have used the files of Table 7-1.

Name	*Length (ms)*	*Ch.*	*Bits/ sample*	*Format*	*Sampling freq.*
seq-3341-7_seq-3342-5-24bit.wav	**28000**	**2**	**24**	**PCM**	**48KHz**
seq-3341-6-6channels-WAVEEX-16bit.wav	**20000**	**6**	**16**	**PCM**	**48KHz**

Table 7-1: Performance test files

All performance measures presented in this paper have been made in a common desktop machine with following features:

Intel core i3, 3GHz, 4GB RAM, Windows 7 ultimate 64b

In the Table 7-2 we can see the total spent time by the LME to process the test files:

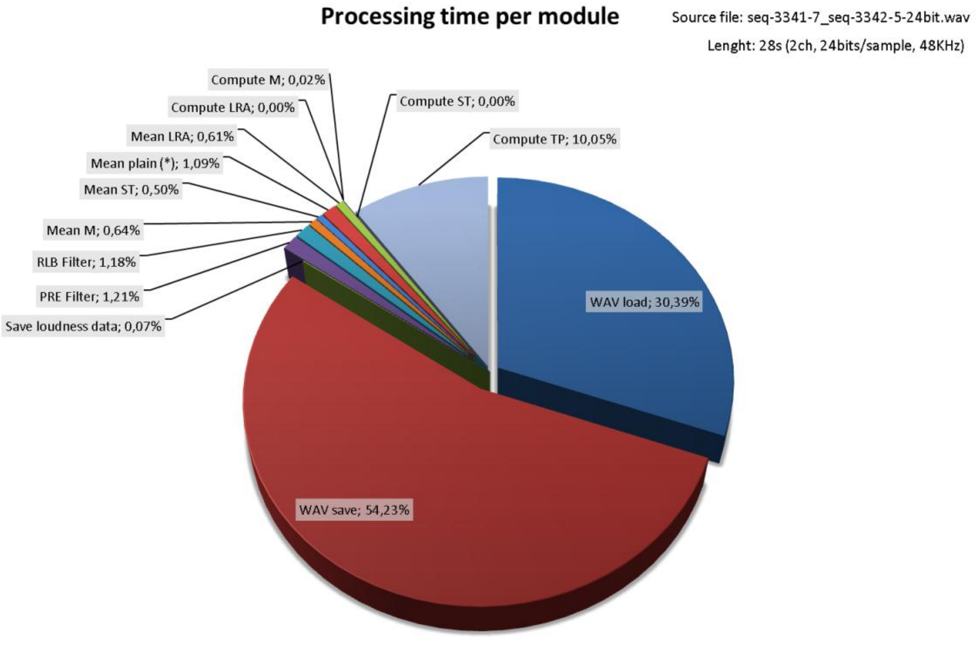
Name	*Length (ms)*	*Process time (ms)*
seq-3341-7_seq-3342-5-24bit.wav	28000	**4228**
seq-3341-6-6channels-WAVEEX-16bit.wav	20000	**7278**

Table 7-2: Performance test results using all modules

As we can see in the previous table both files can be processed faster than real time, but analyzing the detailed log file that generates the LME we can go deeper in the performance analysis.

Dividing the used CPU time by the time used per each LME module we realize that more than 90% of CPU time is consumed by modules that does not involved in loudness computation (WAV read (*), WAV write(*), and true peak), see Figure 7-8.

(*) We have to notice that the WAV read and WAV write modules are not optimized because they are implemented only for test purposes.



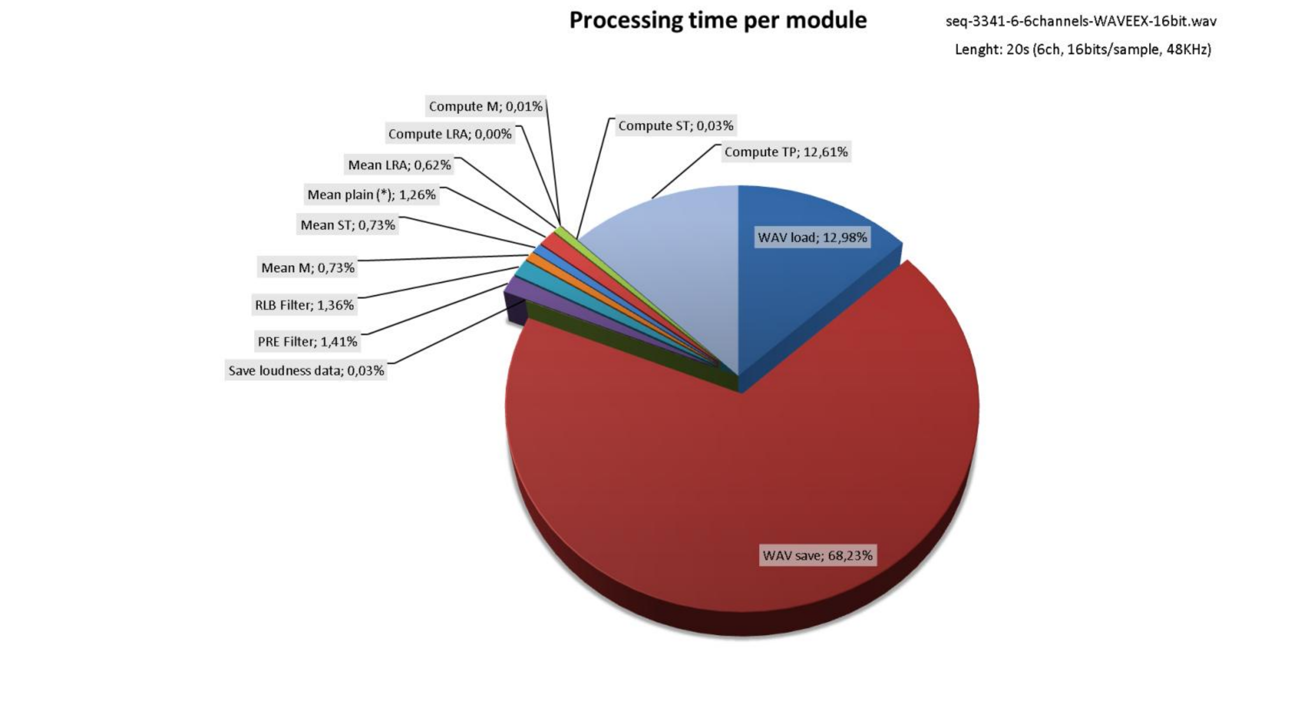


Figure 7-8: Processing time per module

However we want to know the time that LME will need to compute loudness data from audio samples, the "Loudness process time". This will be the loudness reference processing time when the LME runs in real time mode.

To compute the "Loudness process time" we take into account only the modules that are involved in loudness computation: PRE filter, RLB filter, mean M, mean ST, mean plain, compute M (sum channel data and compute log M value), compute ST (sum channel data and compute log ST value). The results are showed in Table 7-3.

Name	***Length (ms)***	***Loudness process time (ms)***
seq-3341-7_seq-3342-5-24bit.wav	28000	**185**
seq-3341-6-6channels-WAVEEX-16bit.wav	20000	**415**

Table 7-3: Performance test results of loudness process modules

And the main conclusions about performance are:

- The loudness LME can analyze audio **files** of 6 audio channels more than 2 times faster than real time in a common desktop machine.

- The LME **can analyze in real time the loudness parameters of 300 audio streams (1 channel per stream) in a common desktop machine** (remember that in Loudness realtime analysis the LME does not use WAV Read and TP modules).

In the annex 15.2 you can find the raw data of this performance analysis.

7.1.2 “EBU mode” compliance

The “EBU Mode” is a measuring mode that loudness meters can include in their set up, when “EBU mode” is activated it means that the meters will comply all requirements mentioned in the documents EBU Tech 3341 [34] and EBU Tech 3342 [35]. These requirements include:

- Measure time scales.
- Integration times.
- Measure gate definition.
- LRA definition and units.
- Display definition.

The “EBU Mode” can be considered as a global preset of the loudness meter.

If a loudness meter wants to be “EBU Mode” compliance it has to pass the compliance test that is announced in the EBU Tech 3341 and EBU Tech 3342. Those tests are composed of 14 audio files of different natures, and the meter has to compute the desired value within the fixed tolerance.

Our loudness meter has passed with high accuracy the “EBU mode” compliance tests

If you want to see all results of the “EBU mode” test you can see the annex 15.3.

Following EBU rules, if we pass the “EBU Mode” compliance test we are able to use the EBU R128 logo (see Figure 7-9) in our meter.

Figure 7-9: EBU R128 logo

7.2 Directshow wrapper

In order to give more flexibility to our LME we have developed a directshow wrapper, this wrapper allows us to measure the loudness data of any type of media file or live signal that can be decoded using directshow technology [42].

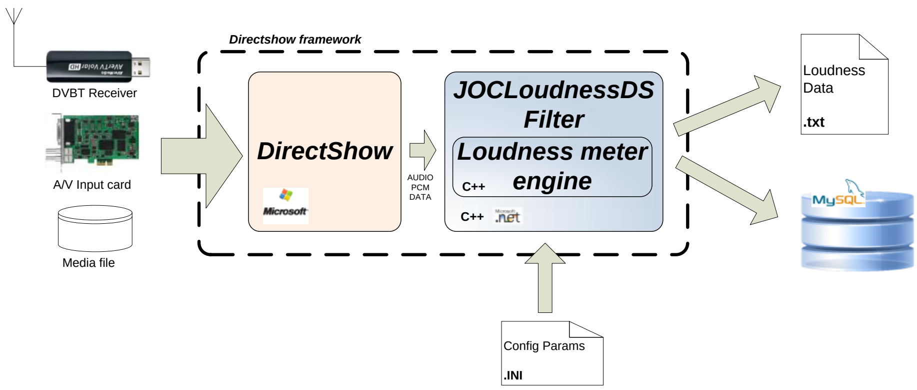


Figure 7-10: Conceptual block diagram of LME directshow wrapper

In brief, directshow is a software framework created by Microsoft designed to handle media files or signals. Its main idea is to use the graph technique to decode or encode media. Every module (named filter) has a determined function (decoder, coder, tuner, file reader, etc…) and it could have many inputs and many outputs in order to connect it to other filters.

In Figure 7-11 we can see a directshow graph that decodes and shows a windows media video/audio file. This graph includes the JOCLoudnessDS, which is the developed wrapper of the LME.

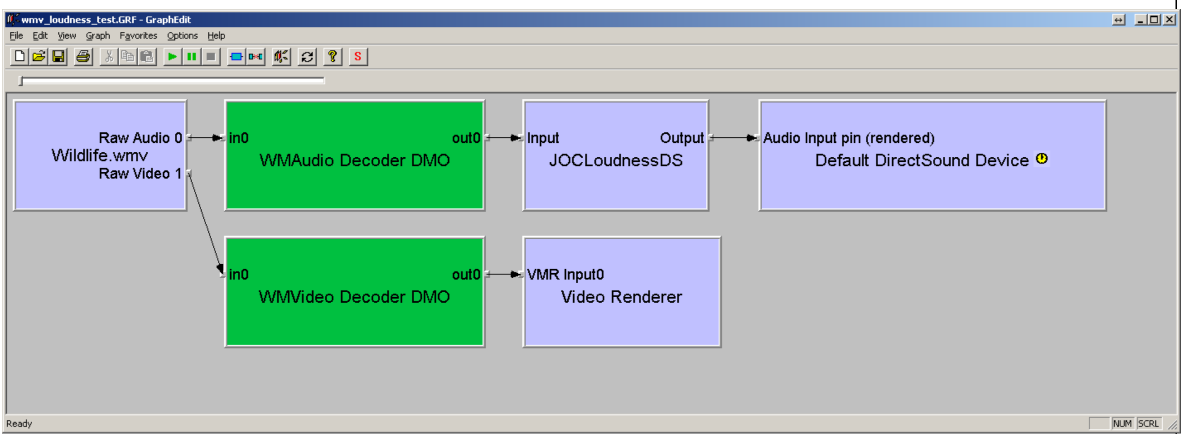


Figure 7-11: Directshow graph

As you can see JOCLoudnessDS has 1 input and 1 output. It processes all samples that enter into it, and after that those samples are sent untouched to the output. The loudness data (Momentary, Short Term and True Peak) can be sent in real time to a file or to a database. In the Figure 7-12 you can see the configuration form of CJOCLoudnessDS filter.

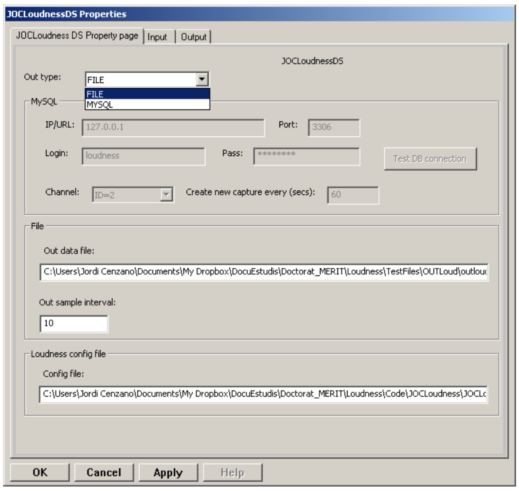


Figure 7-12: CJOCLoudnessDS directshow filter properties form

There exists other similar frameworks that run in different operating systems in which our LME could be integrated like: gstreamer [43], Quicktime [44], FFMpeg [45], etc… Or specialized audio based frameworks as: MAC OS Audio units (AU) [46], or Virtual Studio Technology (VST) plug-ins [47].
All of these frameworks could be used to implement a loudness meter based in our LME due to code portability mentioned in chapter 7.

The reasons because we have chosen the directshow framework to develop our moniroring system are:

- Our systems are windows based.
- It supports natively media files and DTT decoding graphs.
- It is wide extended and exists many codecs developed for it.
- It can work with video and audio.
- Reliable framework (after several years of improving now we can say that it is a reliable framework).

8 Our Loudness monitoring system (LMS)

Based on LME we have developed a complete eco system of applications named loudness monitoring system (LMS).
Our LMS is a user friendly distributed system designed to analyze and report the loudness parameters of the broadcasted content in a multichannel environment. For instance all of this data is available in 2 clicks:

- See the loudness graph of momentary (M), short term (ST), and true peak (TP) of the desired channel time range.
- See the loudness graph and the loudness data (I, LRA, M, ST, TP, and maximum TP) of a singular broadcasted item.
- Sort all broadcasted items by I (from louder to silent).
- See the broadcasted playlist with I, LRA, and maximum TP (maxTP) of each broadcasted event.
- Etc... (See the annex 15.6).

8.1 LMS introduction

The LMS is designed to tune, decode and analyze N broadcast channels at the same time; it can be linked with N broadcast channel playout systems in order to match the loudness data to broadcasted events, this link transforms this monitoring system in a very powerful loudness analysis tool for broadcasters.

We use a database (DB) to store all loudness and system data; you can find the DB definition and description in the annex 15.5.

The most useful modules are the reporting modules (*JOCLoudnessShow*, and *JOCLoudnessASPShow*); these modules can show the loudness data mixed with the asrun data in a large variety of reports as you can see in the following chapters.

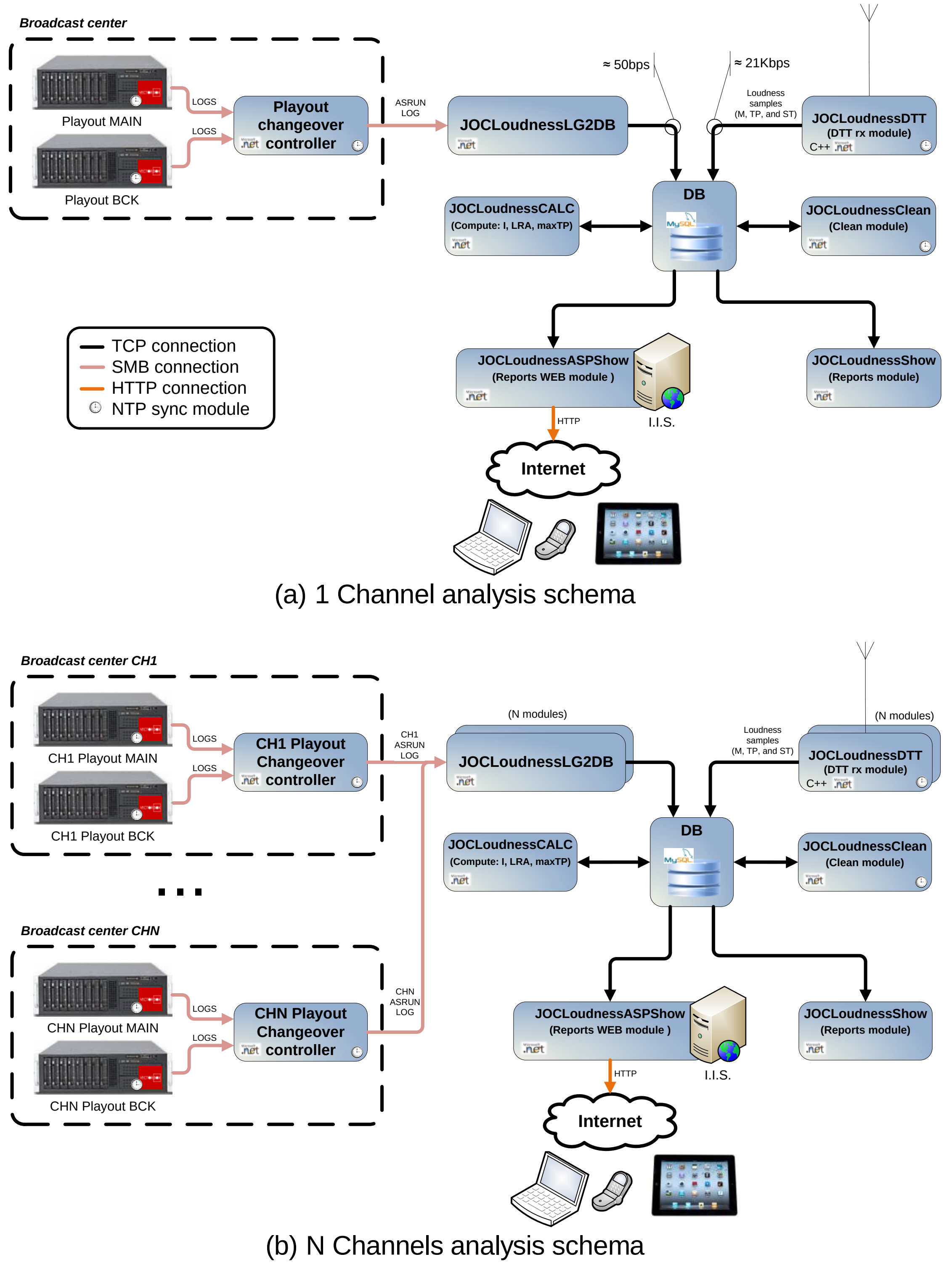


Figure 8-1: Loudness monitoring system block diagram

As we can see in Figure 8-1 there are 2 data types entering into the LMS:

- Asrun log data
  - All professional playout systems generate in real time a precise log file named asrun log. In this log file we can find the exact timestamps created when a programme starts, pauses, and stops, the unique id of the programmes, and much more data. Then the *CH Playout changeover controller* filters the asrun log of the main and backup playouts and only allows the logs of the playout that are on air to pass through.

Finally the *JOCLoudnessLG2DB* translates in real time the asrun log of the redundant playout system into understandable terms for LMS and inserts that data into the DB.

- Loudness data
  - The *JOCLoudnessDTT* tunes and decodes the broadcasted signal of the channel, then computes in real time the M, ST, and TP loudness parameters, and every X seconds it sends these loudness parameters tagged with its timestamps into DB.

The module *JOCLoudnessCALC* continuously asks to the DB for new broadcasted content, and when it is found, it computes its loudness integrated value (I), its loudness range value (LRA), and its Max True Peak value (maxTP). Finally these parameters are stored into the DB.

The *JOCLoudnessCLEAN* is the module that takes care of the DB health, cleaning the oldest data automatically.

The modules *JOCLoudnessShow* and *JOCLoudnessASPShow* are the most complex modules of the system; they are the data mining and reporting modules.
The *JOCLoudnessASPShow* can be accessed with a simple browser (smartphone, tablet, etc…).

In the following sections we will explain the details of every LMS module.

8.2 LMS Architecture

The architecture of the LMS is extremely scalable, all modules are independent software instances that can run in independent machines and they only need to see the database through a specific TCP protocol port (TCP 3306 is the default port for MySQL DB configuration).

The LMS architecture allows natively:

- Multichannel: Analyze different broadcast channels at the same time.

- Multiclient: Allows different clients analyzing different loudness data at the same time.

- Fault tolerance: You can install the central DB in a cluster and use different modules and paths per channel.

- Flexibility: All modules can be hot stopped, hot removed, and hot added.

- Virtualization: It is easy to run the LMS into a virtualized environment because all modules are designed to allow this feature.

We have used C++ and .NET technology to implement all modules of LMS solution. We choose .NET because it is a simple programing language with professional capabilities: Integrated IP communication libraries, native multithreading, perfect integration with windows OS, MySQL driver available, etc…
Summarizing, .NET is an ideal environment to develop windows applications, and it has good integration with MySQL and directshow products.
But all of these modules could be rewritten using other programing language, remember that the LME has been written in C++ and its code is portable.

To develop our LMS we have used MySQL as central DB because it is reliable and free DB product that can run on windows, Linux, and MAC OS¡.

8.3 Implementation issues

In this chapter we describe the most important issues that we have found implementing the LMS.

8.3.1 MySQL C++ Connector BUG

We have used the following MySQL connectors in order to insert and query data to MySQL DB:

- MySQL C++ Connector 1.1.0
- MySQL .NET Connector 6.4.4

We have found a bug between MySQL C++ Connector 1.1.0 and Microsoft C++ runtime libraries for Visual Studio 2010 that produces an application crash every time that we try to get a string from DB.
We have solved this issue using only the C++ connector in our Directshow wrapper (7.2) that is the only module that does not need to get strings from database. All other modules have been implemented using .NET connector.

8.3.2 LMS time synch

When we speak about video timing it is important to know that in broadcast video installations all video equipment is synchronized using a determined signal named genlock (could be black burst or tri level). This signal indicates when every video frame (or field) has to be generated, displayed, processed, etc…

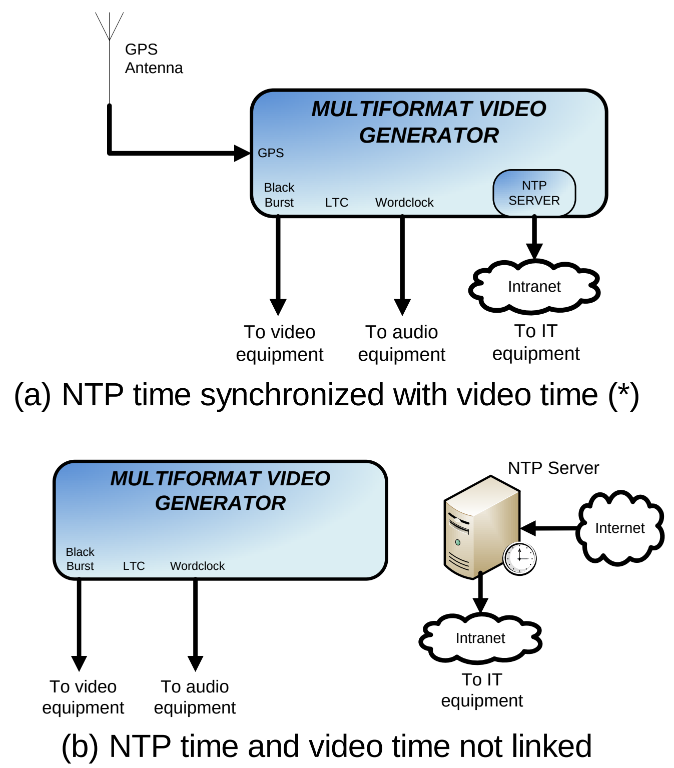


Figure 8-2: Time synchronization models in broadcast installations

(*) Real equipment, see [48]

In the previous figure we can see two different sync models, in the Figure 8-2 (a) an ideal modern installation is shown, the computer time (NTP) is perfectly synchronized to the video time. In this kind of installation we can say that 25 video frames (in PAL) are exactly 1 second in NTP time without any drift. And the equation *8-1* could be used to address to recorded audio samples.

$$Audio(to + ti) = \mathrm{AudioSample}[\mathrm{Int} \quad \mathrm{fs} * \mathrm{ti} \ ] \qquad 8\text{-}1$$

Where **Int()** = Gives us the integer value, **fs** = sampling frequency [samples/s],
ti = NTP time of the sample since capture started [s], **t0 =** NTP capture start time [s]

The Figure 8-2 (b) shows the synchronization model of the most common broadcast video installations where the computer time (NTP) is NOT linked to the video time. In this kind of installations 1 second of NTP time is NOT EXACTLY 25 video frames (in PAL); my experience says that the time drift could be between less than 1s to 20s per day depending on the quality of the video sync generator.

This time drift issue has heavy implications in the LMS because if we want to get from DB the audio sample that are recorded in specific time (*ti*), we **cannot** address it using the equation *8-1*. Knowing that the audio frequency sampling (*fs*) is always linked to the gunlock, if we only save the capture NTP start time (*t0*) and we try to get the audio samples of a determined time since capture starts (*ti*) using equation *8-1* we will make a mistake, the misaligned error will be proportional to the amount of *ti*.

To avoid the mentioned time drift problem we have found 2 solutions:

- Link the genlock generator to NTP time using GPS time signal for example (Figure 8-2 (a)).

- Save the audio samples into DB by groups, labeling every group with the NTP time of the first sample. Using this solution the audio sample addressing error will be limited by the time length of the group of audio samples.

In LMS we have implemented the second solution (save audio samples by groups); we decided to implement this option because it was the cheapest one. The default length of the group of samples is 60 seconds.

8.3.3 Microsoft Broadcast Driver Architecture stability

The Microsoft Broadcast Driver Architecture (BDA) is a very good and cheap technology that we can use for testing purposes, however is not stable enough to use it in professional video solutions that have to work in 24x7.

For 24x7 professional DTT tuning systems I recommend the Dektec products [49]. They also have an SDK to integrate them into any solution that you can design.

8.4 LMS core modules

All LMS core modules are implemented using C# programming language, and they can run in any windows machine (XP, 7, 2003 server, 2008 server, and 32b or 64b).

All core modules have a log file that is automatically generated into the follow directory:

- **[APP PATH]\YYYYMMDD_[name of the moudule].log**

Every day they create a single log file with a detail of all performed actions, encountered problems, warnings, etc…
These log information could be very useful to analyze the performance of the system or in a forensics analysis.

```
19:52:45 22/09/2012 - Action: INFO; Channel:; BroadID:; Message:Application initiated
19:52:46 22/09/2012 - Action: INFO; Channel:; BroadID:; Message:DB Connected
19:52:47 22/09/2012 - Action: INFO; Channel:; BroadID:; Message:Thread (15) - Initializing
19:52:47 22/09/2012 - Action: INFO; Channel:; BroadID:; Message:Thread (15) - connected to DB
19:52:47 22/09/2012 - Action: INFO; Channel:; BroadID:; Message:Thread (15) - Loudness preset loaded =
PRESET_MODE_R128_EU
19:52:47 22/09/2012 - Action: PROCESSING; Channel:RAC105tv - BCN; BroadID:TV01669; DateStart:00:05:53
19/08/2012; Duration:00:03:15; Message:Processing
19:52:47 22/09/2012 - Action: ERROR; Channel:; BroadID:; Message:No captures found using in out date and
channel conditions
```

Figure 8-3: Log file sample of CALC module

All core modules have the DB parameters section in their main form, see Figure 8-4.

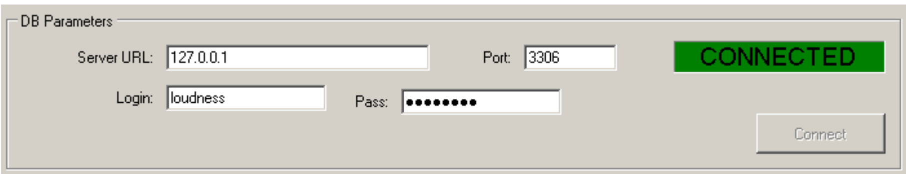


Figure 8-4: DB parameters section

As you can see, to connect to the central DB the core modules only use a single configurable TCP port, this feature allows to the core modules to run in any machine. It is not necessary for all core modules to run into the same computer nor is it necessary that they are in the same network. The only requirement that they need to accomplish is the following simple network setup:

- TCP X port visibility to DB computer (core module acts as client)

X = 3306 by default.

The login and password parameters are the DB user authentication (by default login = loudness, and pass = loudness).

8.4.1 Log to DB Module

The function of this module is to scan the selected directory and when it detects a new asrun file in the monitoring directory it reads the file and it sends the broadcasted events data to DB.
After an asrun file is processed that file is moved to one of the following directories depending on the processing result:

- If file was processed properly: **[Scanning dir]\Processed**
- If an error was found processing the file: **[Scanning dir]\Error**

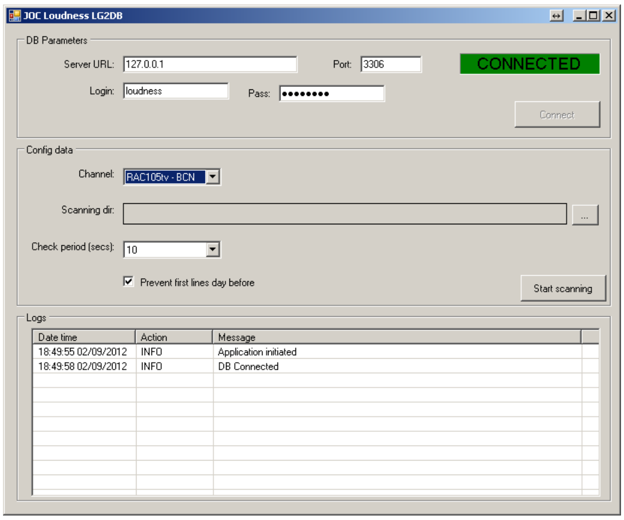


Figure 8-5: Log to DB GUI

This module implements a smart algorithm that can correct some problems in the first and last events of the asrun file: “Prevent first lines day before”.

In the annex 15.6.1 you can find a detailed user manual of this module. Also in the annex 15.6.2 there is a syntax description of a compliant asrun file.

8.4.2 DTT Module

This module is one of the most important core modules; it analyses and sends to DB (in real time) the following loudness parameters of the tuned Digital Terrestrial Television (DTT) channel:

- Momentary value (M)
- Short term value (ST)
- True Peak value (TP)

To set the loudness engine configuration parameters you can use general presets R128 and A/85 or you can load your personalized file preset (see 15.1.3).

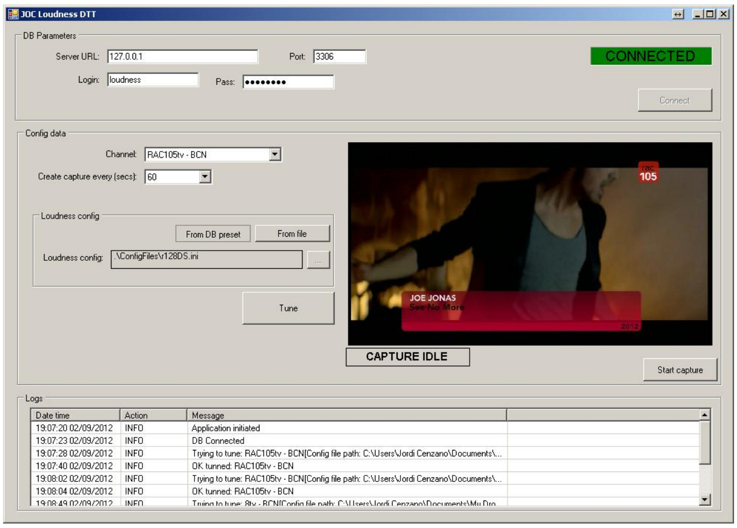


Figure 8-6: DTT module GUI

Internally the DTT module uses directshow technology [42] to tune and decode the selected DTT channel.
In the Figure 8-7 you can see a graphical view of the DTT decoding chain used by this module, The JOCLoudnessDS filter (see chapter 7.2) is added in the audio decoding chain in order to analyze the audio and compute the loudness parameters.

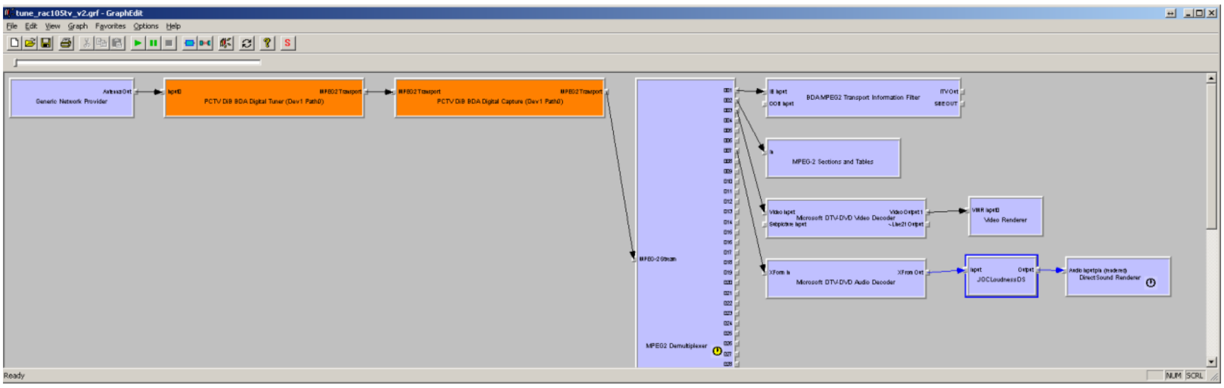


Figure 8-7: Underlying directshow graph

To be more concrete, this module uses directshow Broadcast Driver Architecture (BDA) [50] to tune a DTT channel, this allow us to use any DTT tuning card compliant with BDA technology.
To get our measurements we have used a cheap (40€) PCTV DVB-T Flash Stick 280, see Figure 8-8.

Figure 8-8: Used PCTV DVB-T Flash Stick 280

In the annex 15.6.3 you can find a detailed user manual of this module.

8.4.3 CALC Module

The function of this module is to join the asrun data information with the captured loudness raw data and compute the follow loudness parameters that belong to every broadcasted item:

- Integrated loudness value (I).
- Loudness range (LRA).
- Maximun True Peak value (maxTP).

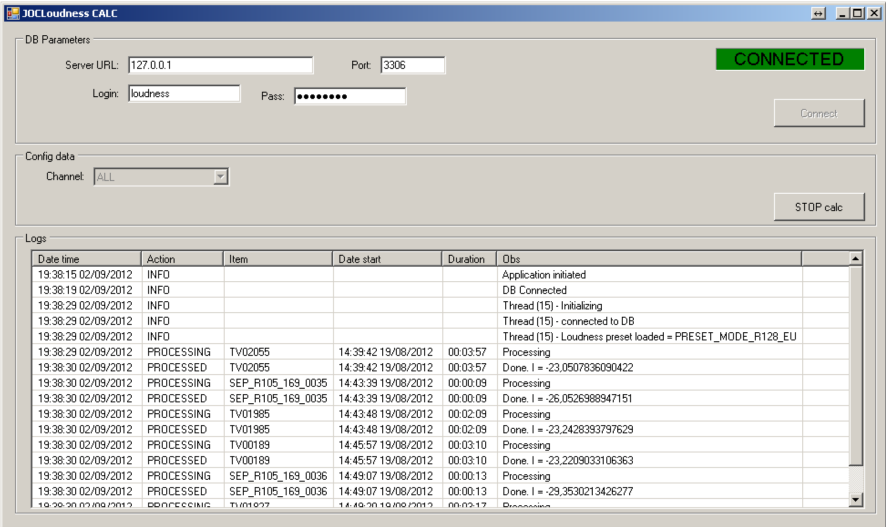


Figure 8-9: CALC module GUI

In the Figure 8-10 you can see the CALC module detailed algorithm. In brief, it looks for broadcasted items with uncomputed loudness parameters, and when it finds one it computes its loudness parameters and it updates its state.

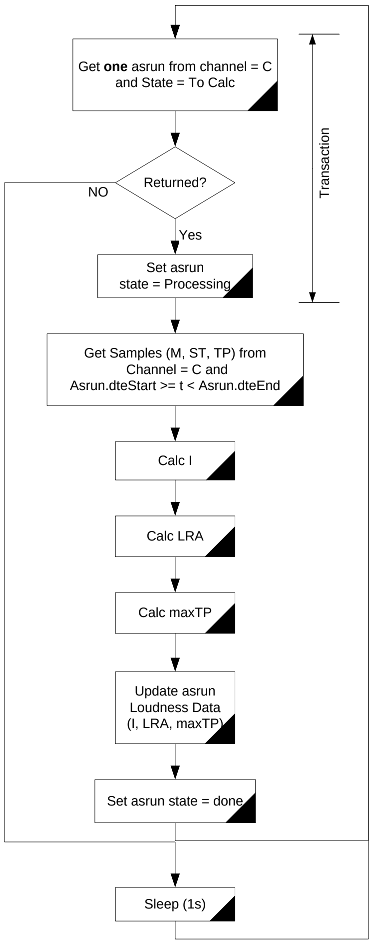


Figure 8-10: CALC module algorithm

You can open any instances of this module as you wish, either on in the same computer or in different computers.
You can configure each instance of the CALC module to scan items broadcasted by one determined channel or to scan all broadcasted items that are in DB, depending on the prioritization criteria: One channel priority or increase global process speed.

In the annex 15.6.4 you can find a detailed user manual of this module.

8.4.4 SHOW Module

The main function of this module is to group and show in an understandable way all data collected by the other modules of the LMS (loudness and asrun).

The show module also have an export function in order to export any loudness data that can appear in a graph or in a report to a text file, this feature allows you to analyze the loudness data with any other data mining solution or import it in a numerical computing environment like matlab or similar, see chapters 15.6.5, and 15.7.

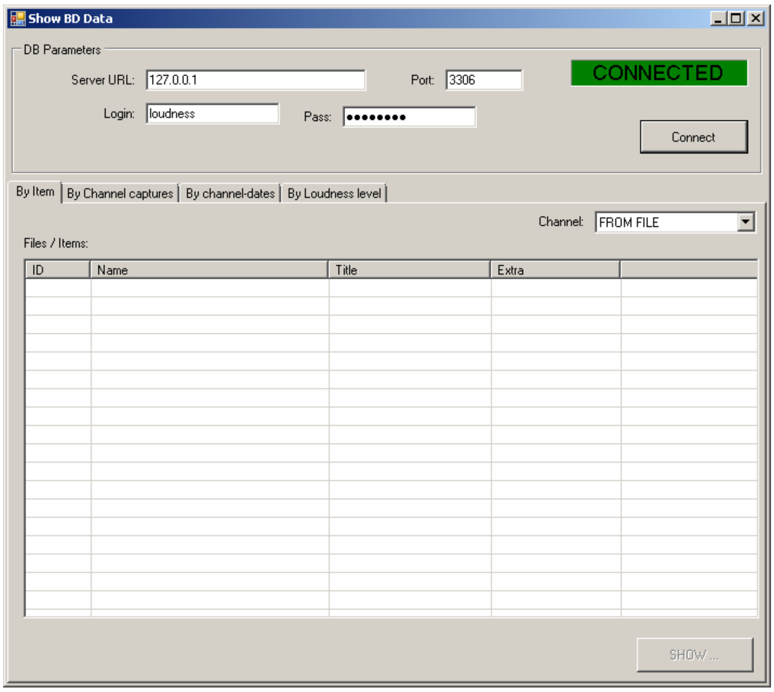
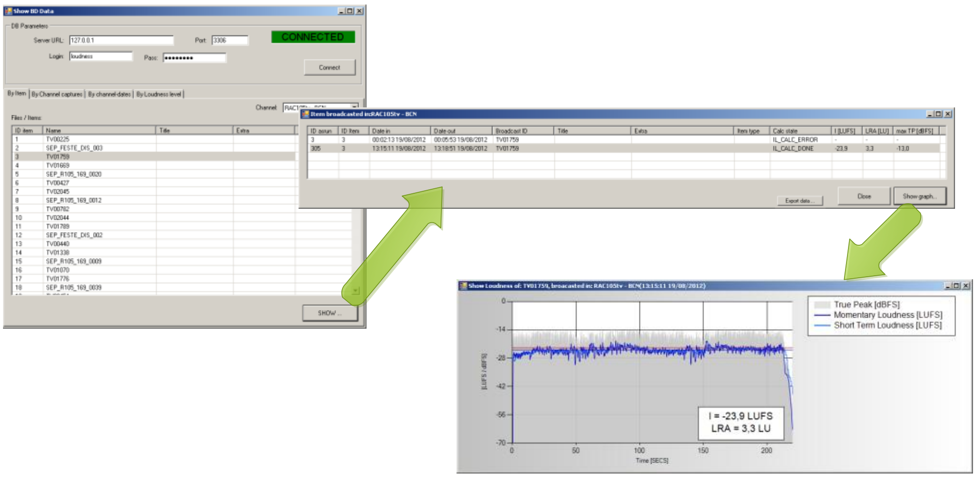


Figure 8-11: SHOW module GUI

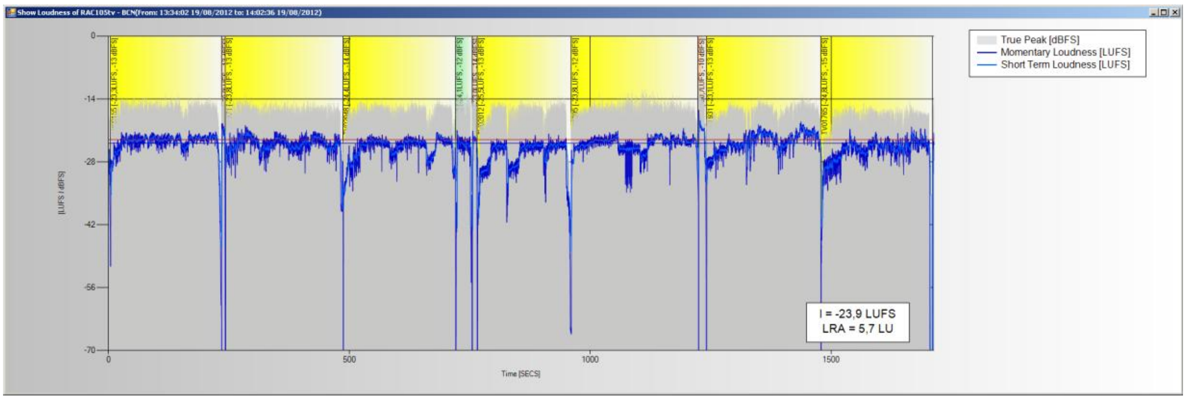
With this module you can create in easy way reports of loudness based on:

- One broadcasted event (*), see Figure 8-12a.
- Data range mixed with asrun data, see Figure 8-12b.
- List of broadcasted contents ordered by loudness values (I, or LRA, or maxTP), see Figure 8-12c.
- Query a list of contents between a loudness integrated (I) range, see Figure 8-12c.

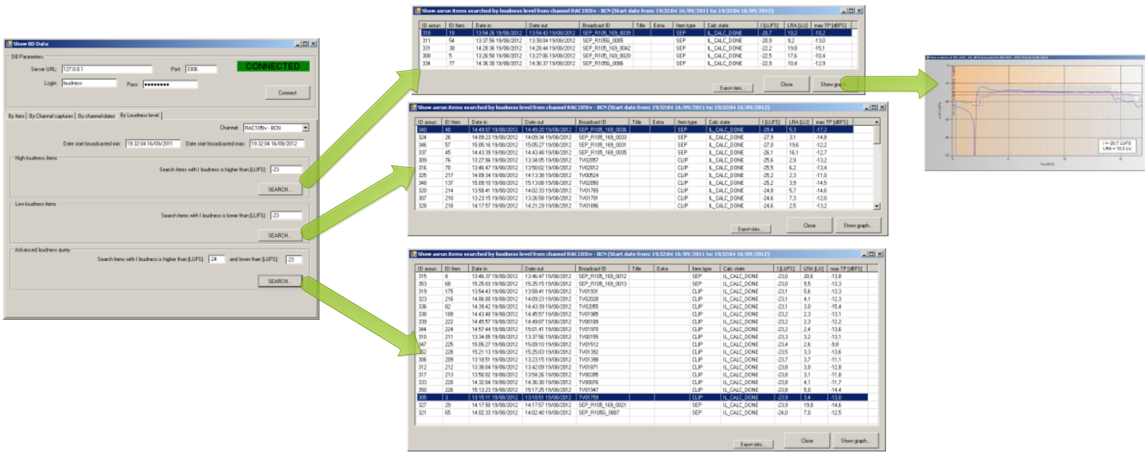
(*) A single broadcast item could be broadcasted many times, for instance an ad could be broadcasted 10 times per day but the broadcast item is the same (same ID code). We define a broadcast event as every time that we broadcast this ad.



(a) Broadcasted event loudness graph



(b) Data range loudness graph mixed with asrun data



(c) Loudness reports by integrated (I) value

Figure 8-12: Different loudness graphs

You can create any graphs that you wish and change from one to another, zoom in and out. In the Figure 8-13 you can see an introduction into the loudness graphs but in the annex 15.7 you can find a complete guide to operate with them.

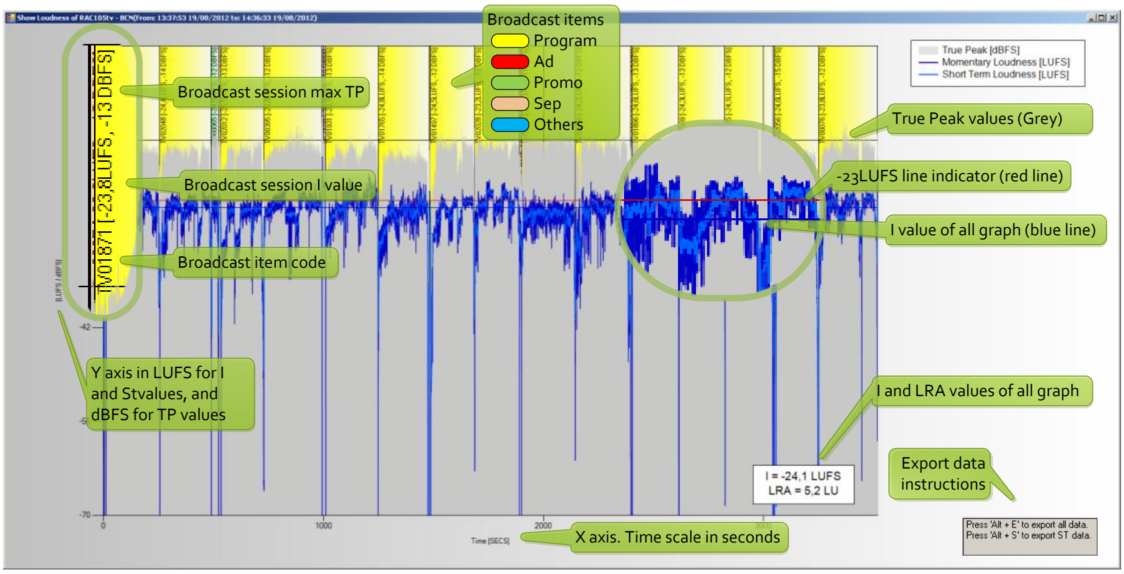


Figure 8-13: Introduction into loudness graphs

In the annex 15.6.5 you can find a user manual of this module with a detailed explanation about how to use the report generator.

8.4.5 CLEAN Module

This module prevents the system degradation by automating the old data deletion.

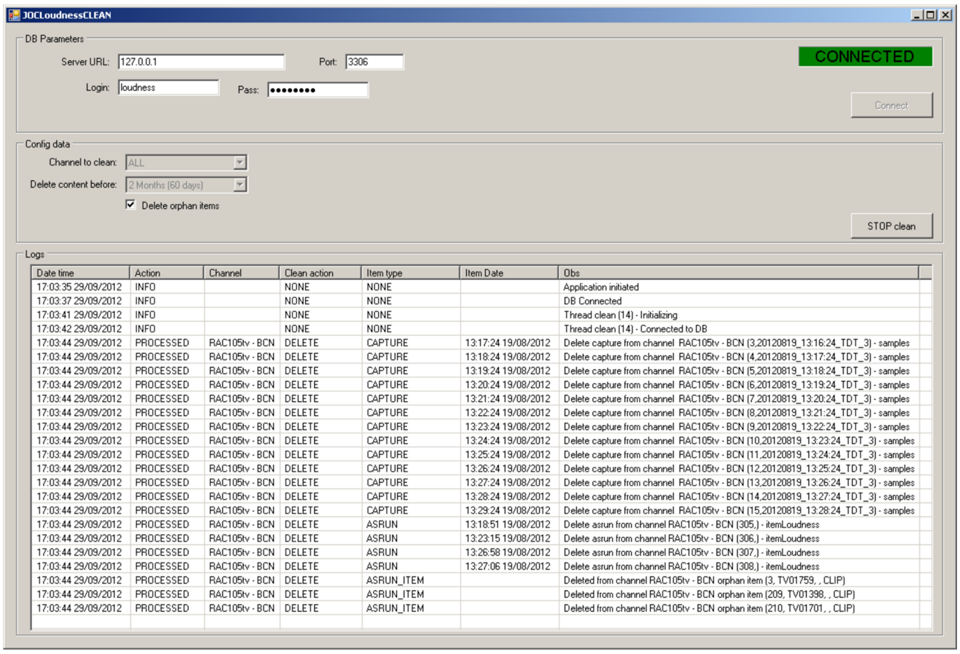


Figure 8-14: CLEAN module GUI

Once a channel and a date threshold are selected this module automatically deletes the data that is older than the threshold, which includes:

- Captures and its related samples
- Asrun items and its related loudness data
- Orphan items (orphan items are the items that aren't related with any asrun entry).

In the Figure 8-15 you can see algorithm of this module.

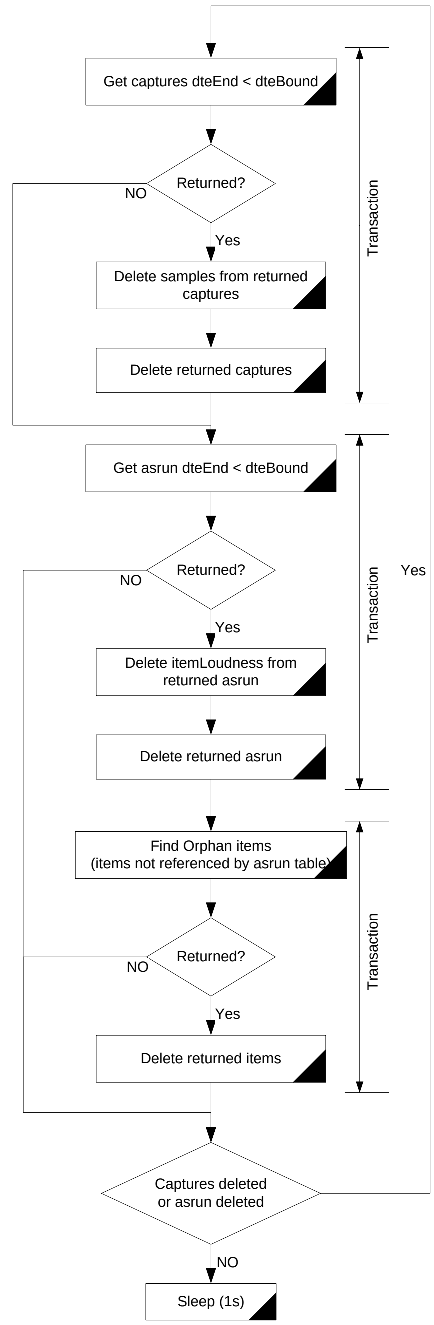


Figure 8-15: CLEAN module algorithm

In the annex 15.6.6 you can find a user manual of this module.

8.5 LMS satellite modules

There are 2 modules that do not belong to the LMS core but are useful: Playout changeover, and ASP SHOW.

8.5.1 Playout change over controller module

This module is strongly linked to the implemented channel playout solution; every playout solution has its own asrun log syntax, and its own redundancy system.
The main task of this module is to convert the asrun log generated by a redundant playout system into an understandable asrun log for LMS.

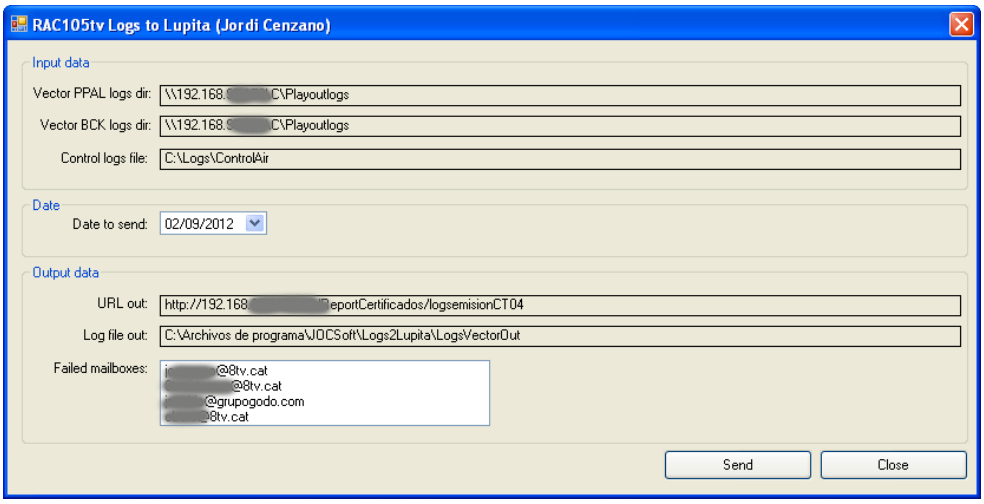


Figure 8-16: Playout module GUI

For every different playout system this module has to be customized. For this paper we have adapted this module to work with VectorBox playout system [51], and MCON playout system [52].

8.5.2 ASP SHOW Module

The ASP SHOW module is a small module developed in ASP language that allows us to show loudness graphs by date (see 15.6.5.3) using a simple web browser as you can see in Figure 8-17.

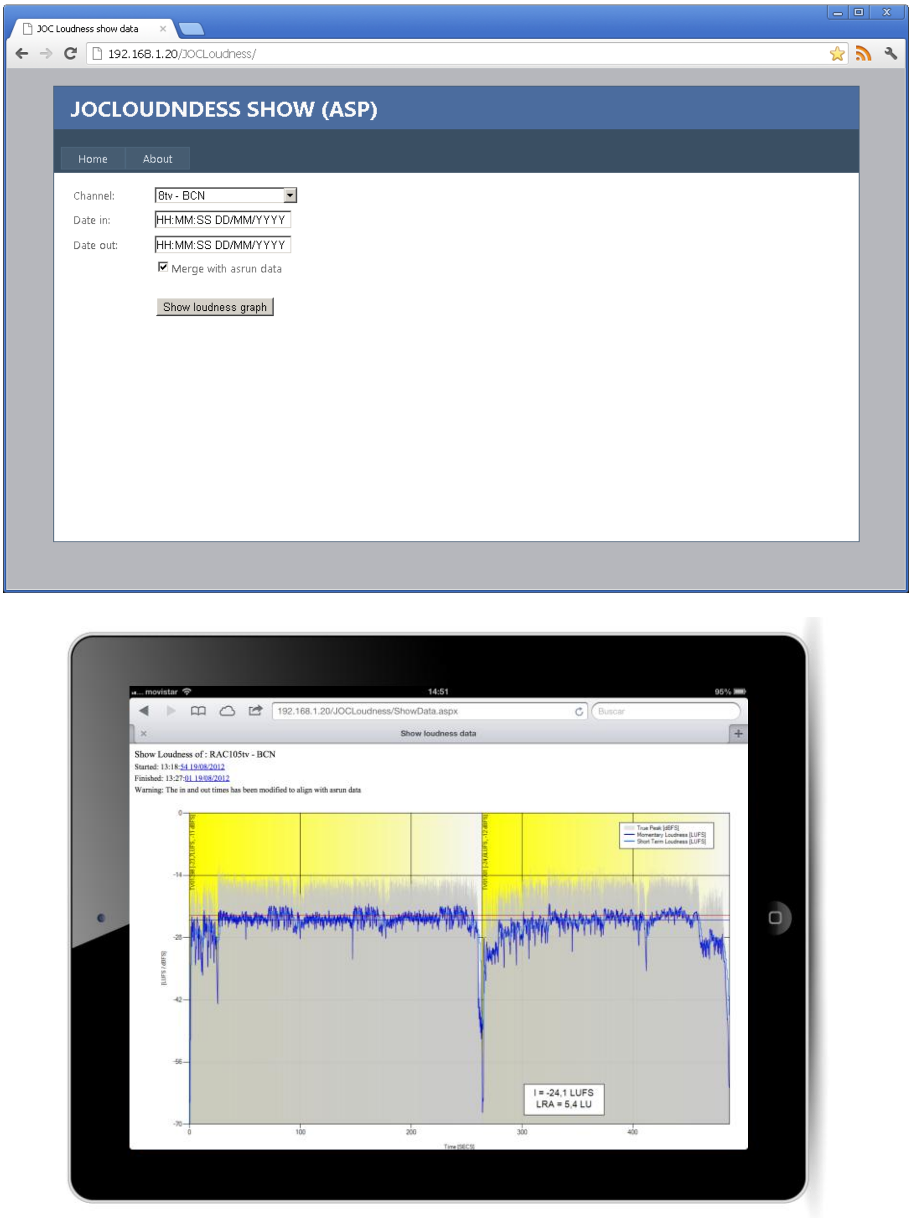


Figure 8-17: ASP SHOW module example

Using this feature you can see a loudness graph of your channel from everywhere, and with any device that can run a simple web browser (laptop, tablet, smartphone, etc…).

9 Example measures

In order to test the system we have made some real measurements to three Spanish broadcasters (8tv, RAC105tv, TV3).

In the Figure 9-1 we can see the set up used to get the following loudness measures. We have to notice that the **cost of the complete measurement equipment (hardware and software) does not exceed 1.000€.**

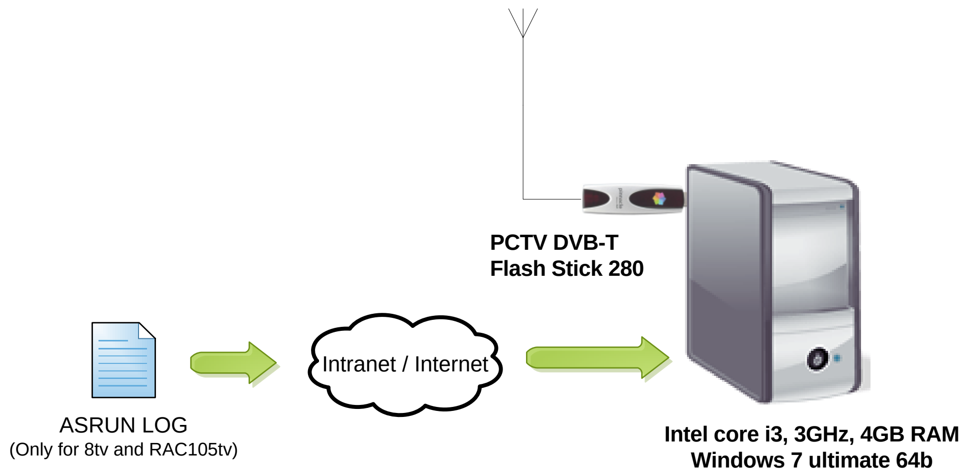


Figure 9-1: Loudness test measures set up

We have to mention that 8tv and RAC105tv have allowed us to access to their asrun information in real time.

In order to compare the measures **<u>we have analyzed 4 hours (from 20h to 24h) of a single day</u>** of the broadcasted content for each channel, and we have used the R128 preset in our LMS.

Note: In Spain the commercials are limited by law to 12min per hour.

9.1 8tv

8tv is a regional broadcaster in Spain. It is a private TV of Catalonia and it broadcasts generalist content (information and entertainment). Its audience (share) it is about 3.4% (from Barlovento report of 03/2013 [53]).

In the following figure we can see the loudness data of 4 hours of different broadcast content mixed with asrun data from 8tv (blue = programs, red = ads). Remember that in the annex 15.7 you can read a complete guide of loudness graphs.

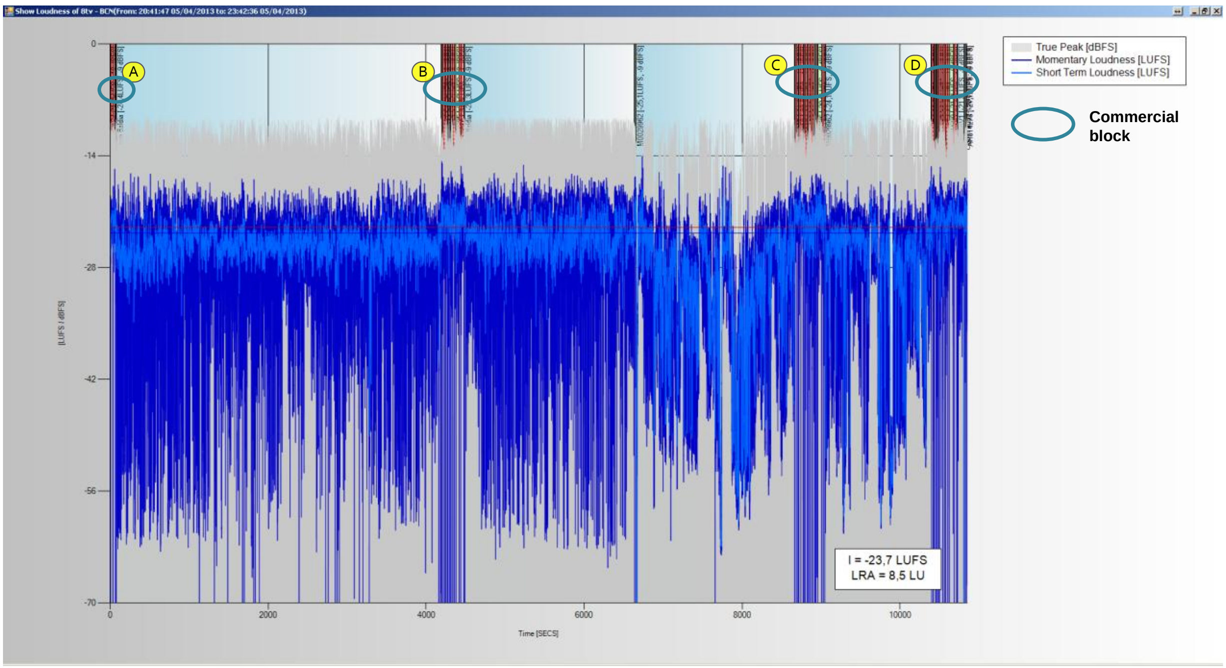


Figure 9-2: 8tv loudness of 05/04/2013 from 20h to 24h

The main conclusions about loudness that you can extract viewing the Figure 9-2 are:

- The 4 hours I mean level loudness (I = -23.7 LUFS) is according with R128.
- The commercial blocks (red parts) are slightly louder that the rest of content. We can deduce that 8tv does not apply a properly R128 normalization.

To analyze with detail the commercial blocks in the Figure 9-3 we can see a zoom in the commercial block B.

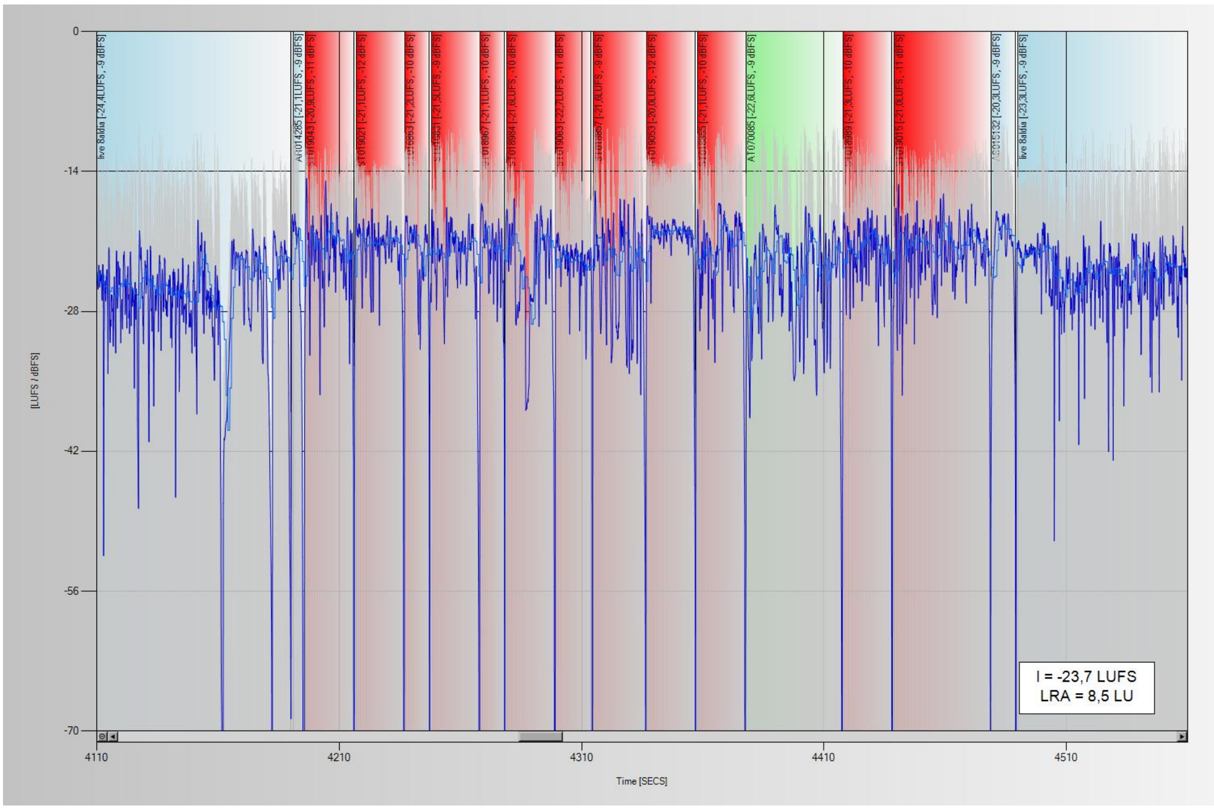
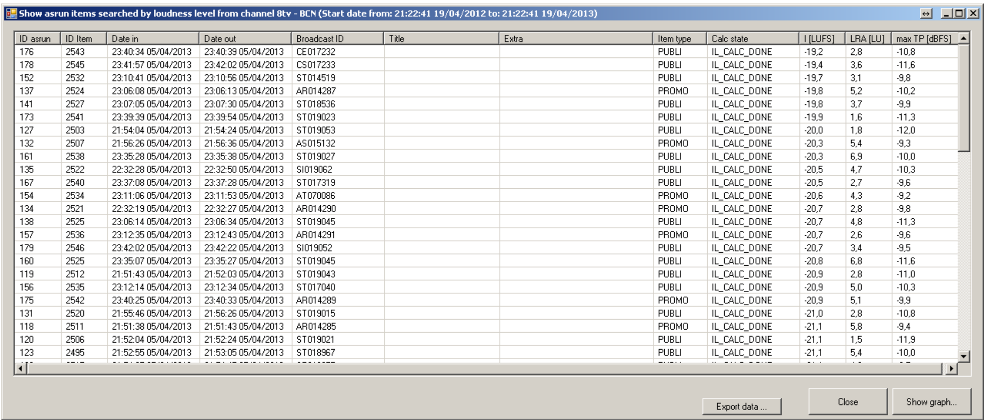


Figure 9-3: 8tv Loudness of commercial block B

By viewing the previous figure we can assert the following:

- The ads use to have higher I loudness than the other content
- The ads use to have a small dynamic range (concepts explained in 3.2.5).
  Note:The silences that appear periodically between ads are the silence gaps that the broadcasters put to separate each other ad.

Using the report "By Loudness Level" of SHOW LMS module we ordered the registered 8tv broadcasted content from louder to quieter, and the result was that the majority of louder content was commercials, see Figure 9-4.



Show asrun items searched by loudness level from channel 8tv - BCN (Start date from: 21:22:41 19/04/2012 to: 21:22:41 19/04/2013)

ID asrun	ID Item	Date in	Date out	Broadcast ID	Title	Extra	Item type	Calc state	I [LUFS]	LRA [LU]	max TP [dBFS]
176	2543	23:40:34 05/04/2013	23:40:39 05/04/2013	CE017232			PUBLI	IL_CALC_DONE	-19,2	2,8	-10,8
178	2545	23:41:57 05/04/2013	23:42:02 05/04/2013	CS017233			PUBLI	IL_CALC_DONE	-19,4	3,6	-11,6
152	2532	23:10:41 05/04/2013	23:10:56 05/04/2013	ST014519			PUBLI	IL_CALC_DONE	-19,7	3,1	-9,8
137	2524	23:06:08 05/04/2013	23:06:13 05/04/2013	AR014287			PROMO	IL_CALC_DONE	-19,8	5,2	-10,2
141	2527	23:07:05 05/04/2013	23:07:30 05/04/2013	ST018536			PUBLI	IL_CALC_DONE	-19,8	3,7	-9,9
173	2541	23:39:39 05/04/2013	23:39:54 05/04/2013	ST019023			PUBLI	IL_CALC_DONE	-19,9	1,6	-11,3
127	2503	21:54:04 05/04/2013	21:54:24 05/04/2013	ST019053			PUBLI	IL_CALC_DONE	-20,0	1,8	-12,0
132	2507	21:56:26 05/04/2013	21:56:36 05/04/2013	AS015132			PROMO	IL_CALC_DONE	-20,3	5,4	-9,3
161	2538	23:35:28 05/04/2013	23:35:38 05/04/2013	ST019027			PUBLI	IL_CALC_DONE	-20,3	6,9	-10,0
135	2522	22:32:28 05/04/2013	22:32:50 05/04/2013	SI019062			PUBLI	IL_CALC_DONE	-20,5	4,7	-10,3
167	2540	23:37:08 05/04/2013	23:37:28 05/04/2013	ST017319			PUBLI	IL_CALC_DONE	-20,5	2,7	-9,6
154	2534	23:11:06 05/04/2013	23:11:53 05/04/2013	AT070086			PROMO	IL_CALC_DONE	-20,6	4,3	-9,2
134	2521	22:32:19 05/04/2013	22:32:27 05/04/2013	AR014290			PROMO	IL_CALC_DONE	-20,7	2,8	-9,8
138	2525	23:06:14 05/04/2013	23:06:34 05/04/2013	ST019045			PUBLI	IL_CALC_DONE	-20,7	4,8	-11,3
157	2536	23:12:35 05/04/2013	23:12:43 05/04/2013	AR014291			PROMO	IL_CALC_DONE	-20,7	2,6	-9,6
179	2546	23:42:02 05/04/2013	23:42:22 05/04/2013	SI019052			PUBLI	IL_CALC_DONE	-20,7	3,4	-9,5
160	2525	23:35:07 05/04/2013	23:35:27 05/04/2013	ST019045			PUBLI	IL_CALC_DONE	-20,8	6,8	-11,6
119	2512	21:51:43 05/04/2013	21:52:03 05/04/2013	ST019043			PUBLI	IL_CALC_DONE	-20,9	2,8	-11,0
156	2535	23:12:14 05/04/2013	23:12:34 05/04/2013	ST017040			PUBLI	IL_CALC_DONE	-20,9	5,0	-10,3
175	2542	23:40:25 05/04/2013	23:40:33 05/04/2013	AR014289			PROMO	IL_CALC_DONE	-20,9	5,1	-9,9
131	2520	21:55:46 05/04/2013	21:56:26 05/04/2013	ST019015			PUBLI	IL_CALC_DONE	-21,0	2,8	-10,8
118	2511	21:51:38 05/04/2013	21:51:43 05/04/2013	AR014285			PROMO	IL_CALC_DONE	-21,1	5,8	-9,4
120	2506	21:52:04 05/04/2013	21:52:24 05/04/2013	ST019021			PUBLI	IL_CALC_DONE	-21,1	1,5	-11,9
123	2495	21:52:55 05/04/2013	21:53:05 05/04/2013	ST018967			PUBLI	IL_CALC_DONE	-21,1	5,4	-10,0

Export data ... Close Show graph...

Figure 9-4: 8tv I Loudness sorted content

9.2 RAC105tv

RAC105tv is a regional broadcaster in Spain. It is a music TV that broadcasts in the Catalonia area. Its audience (share) it is about 0.2% (from Barlovento report of 03/2013 [53]).

In the Figure 9-5 we can see the loudness data of RAC105tv mixed with its asrun data, as we can see the broadcasted content are short music videos with few ads.

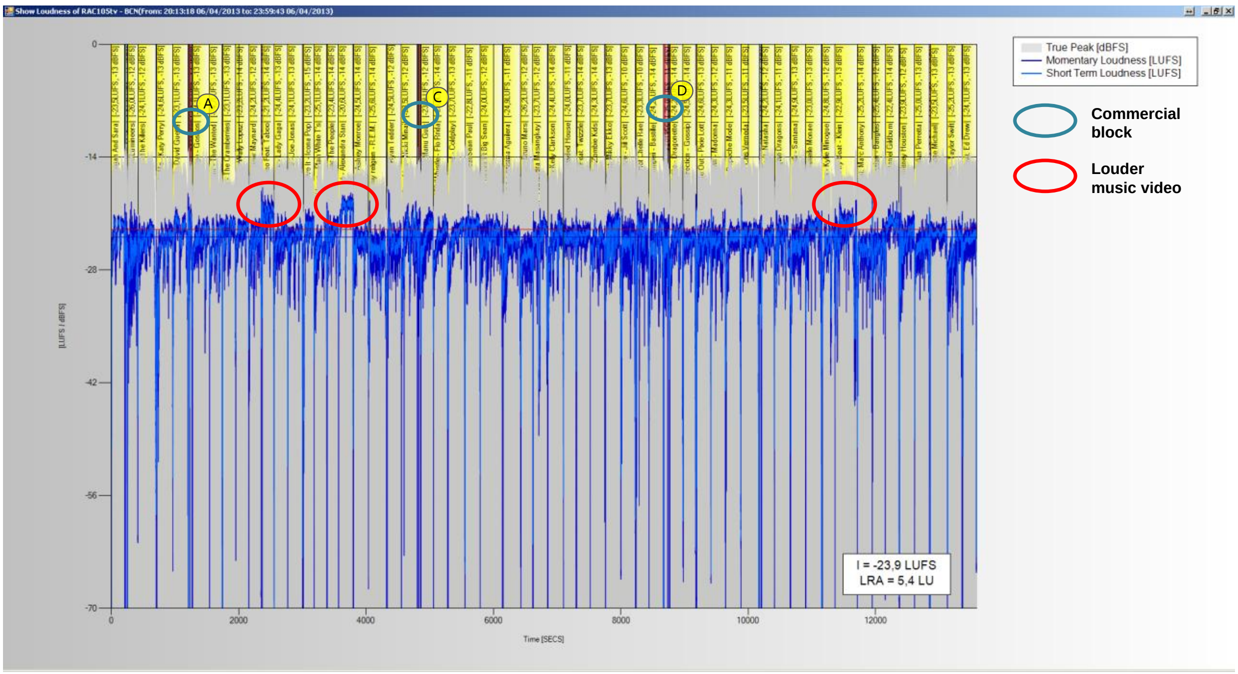


Figure 9-5: RAC105tv loudness of 06/04/2013 from 20h to 24h

By viewing the previous figure we can extract the following conclusions:

- The 4 hours I mean level loudness (I = -23.9 LUFS) is according with R128.

- Some music videos have higher loudness than the mean. They could apply better loudness normalization.

In the Figure 9-6 we can see a zoom in to a louder music video.

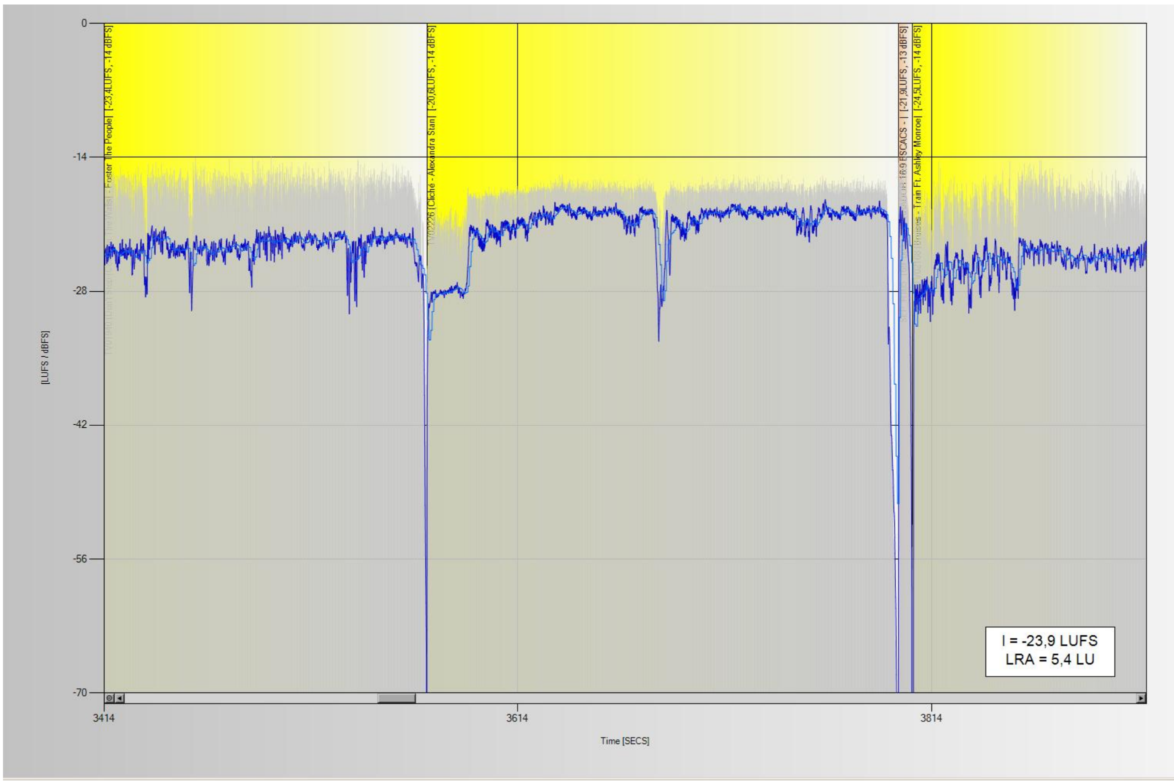
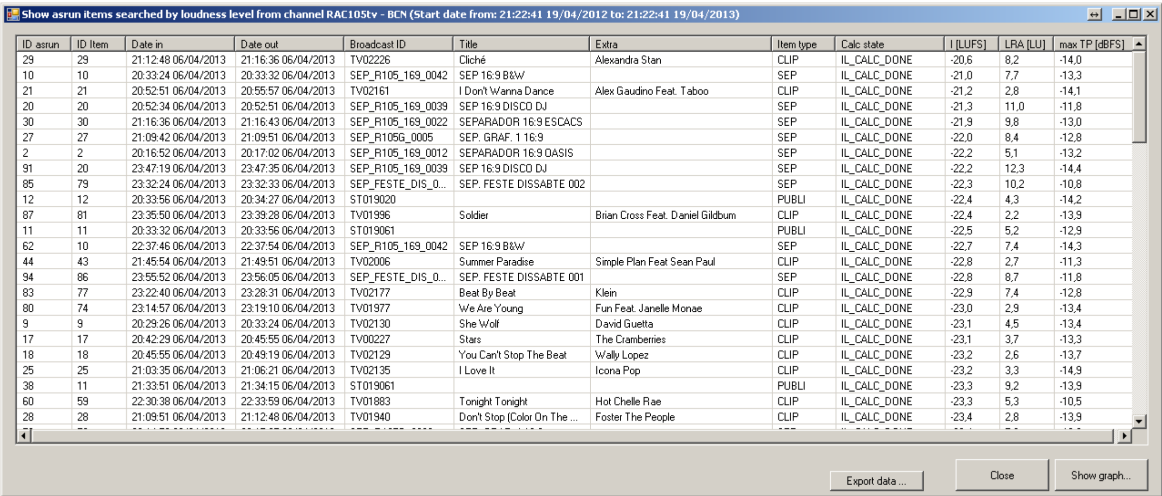


Figure 9-6: RAC105tv high loudness music video

In the following figure we can see the RAC105tv broadcasted content sorted from louder to quieter.



Show asrun items searched by loudness level from channel RAC105tv - BCN (Start date from: 21:22:41 19/04/2012 to: 21:22:41 19/04/2013)

ID asrun	ID Item	Date in	Date out	Broadcast ID	Title	Extra	Item type	Calc state	I [LUFS]	LRA [LU]	max TP [dBFS]
29	29	21:12:48 06/04/2013	21:16:36 06/04/2013	TV02226	Cliché	Alexandra Stan	CLIP	IL_CALC_DONE	-20,6	8,2	-14,0
10	10	20:33:24 06/04/2013	20:33:32 06/04/2013	SEP_R105_169_0042	SEP 16:9 B&W		SEP	IL_CALC_DONE	-21,0	7,7	-13,3
21	21	20:52:51 06/04/2013	20:55:57 06/04/2013	TV02161	I Don't Wanna Dance	Alex Gaudino Feat. Taboo	CLIP	IL_CALC_DONE	-21,2	2,8	-14,1
20	20	20:52:34 06/04/2013	20:52:51 06/04/2013	SEP_R105_169_0039	SEP 16:9 DISCO DJ		SEP	IL_CALC_DONE	-21,3	11,0	-11,8
30	30	21:16:36 06/04/2013	21:16:43 06/04/2013	SEP_R105_169_0022	SEPARADOR 16:9 ESCACS		SEP	IL_CALC_DONE	-21,9	9,8	-13,0
27	27	21:09:42 06/04/2013	21:09:51 06/04/2013	SEP_R105G_0005	SEP. GRAF. 1 16:9		SEP	IL_CALC_DONE	-22,0	8,4	-12,8
2	2	20:16:52 06/04/2013	20:17:02 06/04/2013	SEP_R105_169_0012	SEPARADOR 16:9 OASIS		SEP	IL_CALC_DONE	-22,2	5,1	-13,2
91	20	23:47:19 06/04/2013	23:47:35 06/04/2013	SEP_R105_169_0039	SEP 16:9 DISCO DJ		SEP	IL_CALC_DONE	-22,2	12,3	-14,4
85	79	23:32:24 06/04/2013	23:32:33 06/04/2013	SEP_FESTE_DIS_0...	SEP. FESTE DISSABTE 002		SEP	IL_CALC_DONE	-22,3	10,2	-10,8
12	12	20:33:56 06/04/2013	20:34:27 06/04/2013	ST019020			PUBLI	IL_CALC_DONE	-22,4	4,3	-14,2
87	81	23:35:50 06/04/2013	23:39:28 06/04/2013	TV01996	Soldier	Brian Cross Feat. Daniel Gildbum	CLIP	IL_CALC_DONE	-22,4	2,2	-13,9
11	11	20:33:32 06/04/2013	20:33:56 06/04/2013	ST019061			PUBLI	IL_CALC_DONE	-22,5	5,2	-12,9
62	10	22:37:46 06/04/2013	22:37:54 06/04/2013	SEP_R105_169_0042	SEP 16:9 B&W		SEP	IL_CALC_DONE	-22,7	7,4	-14,3
44	43	21:45:54 06/04/2013	21:49:51 06/04/2013	TV02006	Summer Paradise	Simple Plan Feat Sean Paul	CLIP	IL_CALC_DONE	-22,8	2,7	-11,3
94	86	23:55:52 06/04/2013	23:56:05 06/04/2013	SEP_FESTE_DIS_0...	SEP. FESTE DISSABTE 001		SEP	IL_CALC_DONE	-22,8	8,7	-11,8
83	77	23:22:40 06/04/2013	23:28:31 06/04/2013	TV02177	Beat By Beat	Klein	CLIP	IL_CALC_DONE	-22,9	7,4	-12,8
80	74	23:14:57 06/04/2013	23:19:10 06/04/2013	TV01977	We Are Young	Fun Feat. Janelle Monae	CLIP	IL_CALC_DONE	-23,0	2,9	-13,4
9	9	20:29:26 06/04/2013	20:33:24 06/04/2013	TV02130	She Wolf	David Guetta	CLIP	IL_CALC_DONE	-23,1	4,5	-13,4
17	17	20:42:29 06/04/2013	20:45:55 06/04/2013	TV00227	Stars	The Cramberries	CLIP	IL_CALC_DONE	-23,1	3,7	-13,3
18	18	20:45:55 06/04/2013	20:49:19 06/04/2013	TV02129	You Can't Stop The Beat	Wally Lopez	CLIP	IL_CALC_DONE	-23,2	2,6	-13,7
25	25	21:03:35 06/04/2013	21:06:21 06/04/2013	TV02135	I Love It	Icona Pop	CLIP	IL_CALC_DONE	-23,2	3,3	-14,9
38	11	21:33:51 06/04/2013	21:34:15 06/04/2013	ST019061			PUBLI	IL_CALC_DONE	-23,3	9,2	-13,9
60	59	22:30:38 06/04/2013	22:33:59 06/04/2013	TV01883	Tonight Tonight	Hot Chelle Rae	CLIP	IL_CALC_DONE	-23,3	5,3	-10,5
28	28	21:09:51 06/04/2013	21:12:48 06/04/2013	TV01940	Don't Stop (Color On The ...	Foster The People	CLIP	IL_CALC_DONE	-23,4	2,8	-13,9

Export data ...
Close
Show graph...

Figure 9-7: RAC105tv high loudness music video

In RAC105tv 7 of 10 louder contents are advertising separators.

9.3 TV3

TV3 is a regional broadcaster in Spain. It is a public TV of Catalonia and it broadcasts generalist content (information and entertainment). Its audience (share) it is about 13.3% (from Barlovento report of 03/2013 [53]).

In the following figure we can see the loudness data without asrun information (we cannot access to TV3 asrun information).

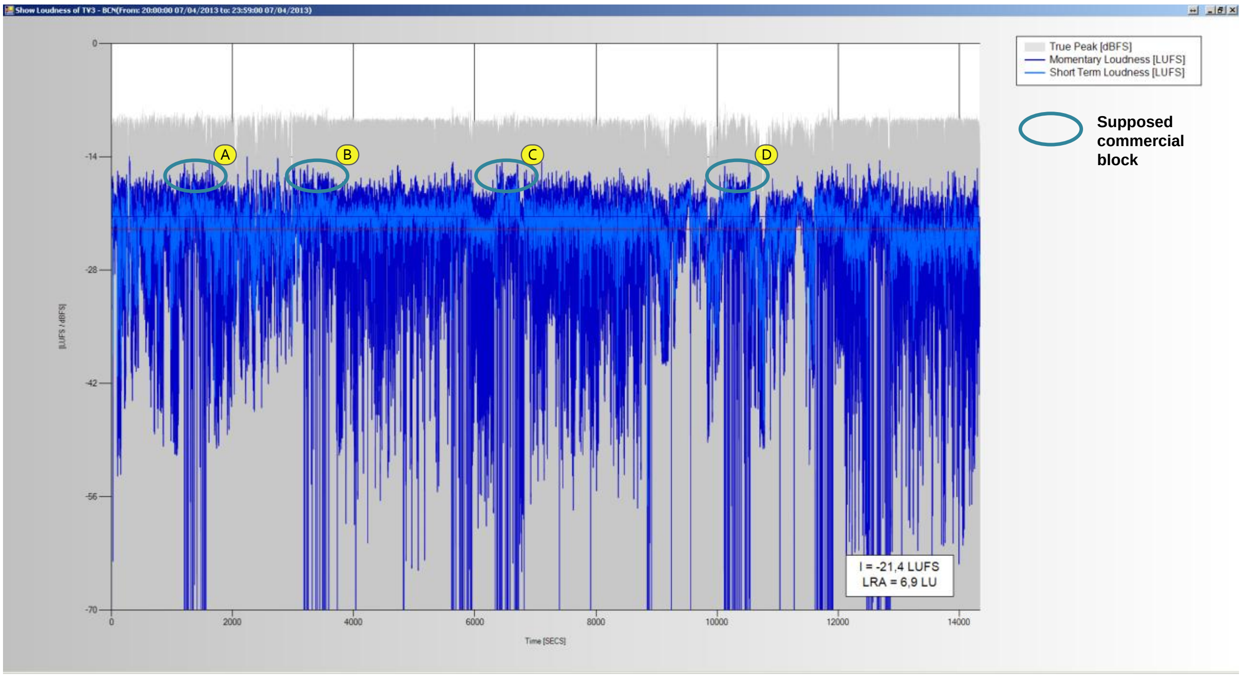


Figure 9-8: TV3 loudness of 07/04/2013 from 20h to 24h

The conclusions that we can deduce are the following (without asrun information are inaccurate):

- The 4 hours I mean level loudness (I = -21.4 LUFS) does not meet R128, it is clearly louder. It is easy to deduce that no loudness normalization is applied in TV3 broadcast chain.
- It seems that the commercial blocks (detected by short separated silences) are slightly louder than the rest of content.

In the following figure we can see a zoom in the supposed commercial block D.

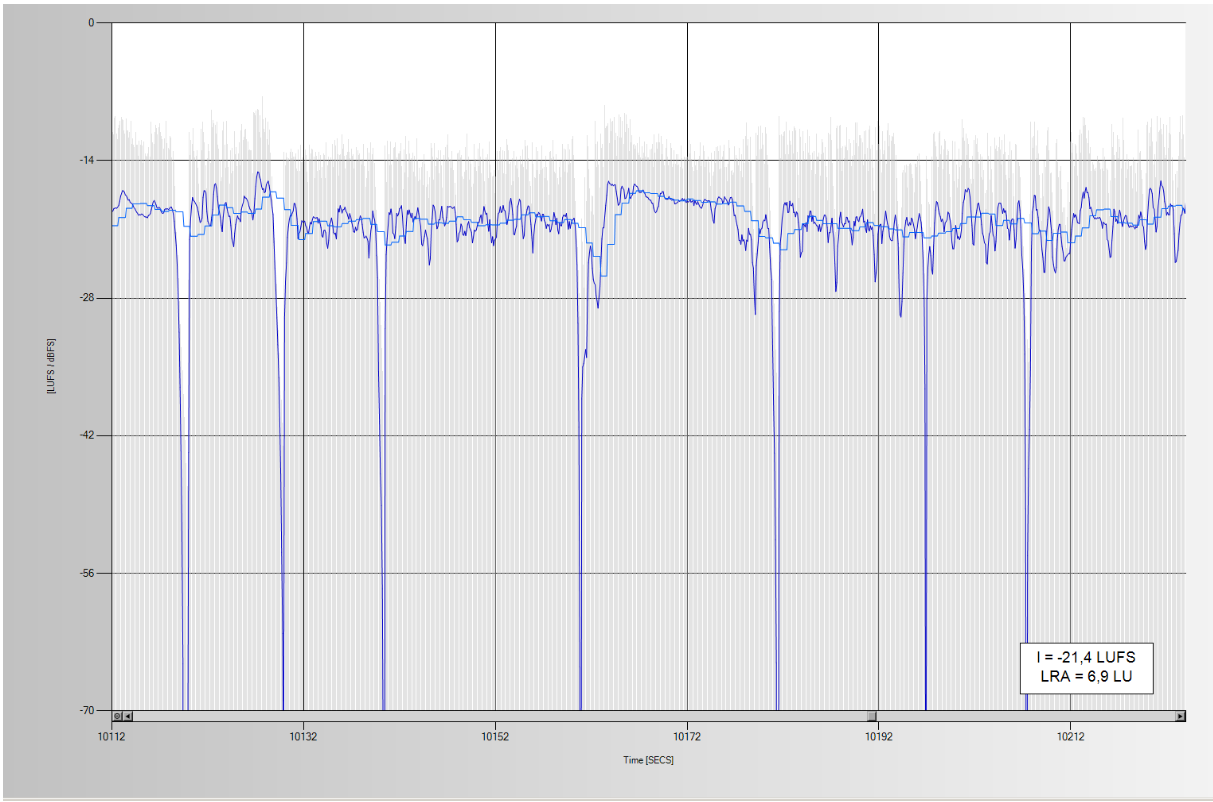


Figure 9-9: TV3 Loudness of commercial block D

9.4 Comparative of channel loudness

In this section we compare the captured loudness data of 3 different Spanish broadcasters. We have to say that a complete loudness analysis is out of the scope of this paper, in this section we will compare only 4 hours of broadcasted content which is not enough to extract solid conclusions.

Note: The following graphics have been done using Microsoft Excel. To extract the loudness data from LMS we have used the LMS SHOW module export functions (see 15.6.5 and 15.7.4).

9.4.1 Per channel

In the Figure 9-10 we can see (from left to right)(*): the integer loudness, the mean of momentary loudness, the mean of short term loudness, the mean of true peak value, the maximum of true peak value.

(*) All of these values are referred to a 4h capture.

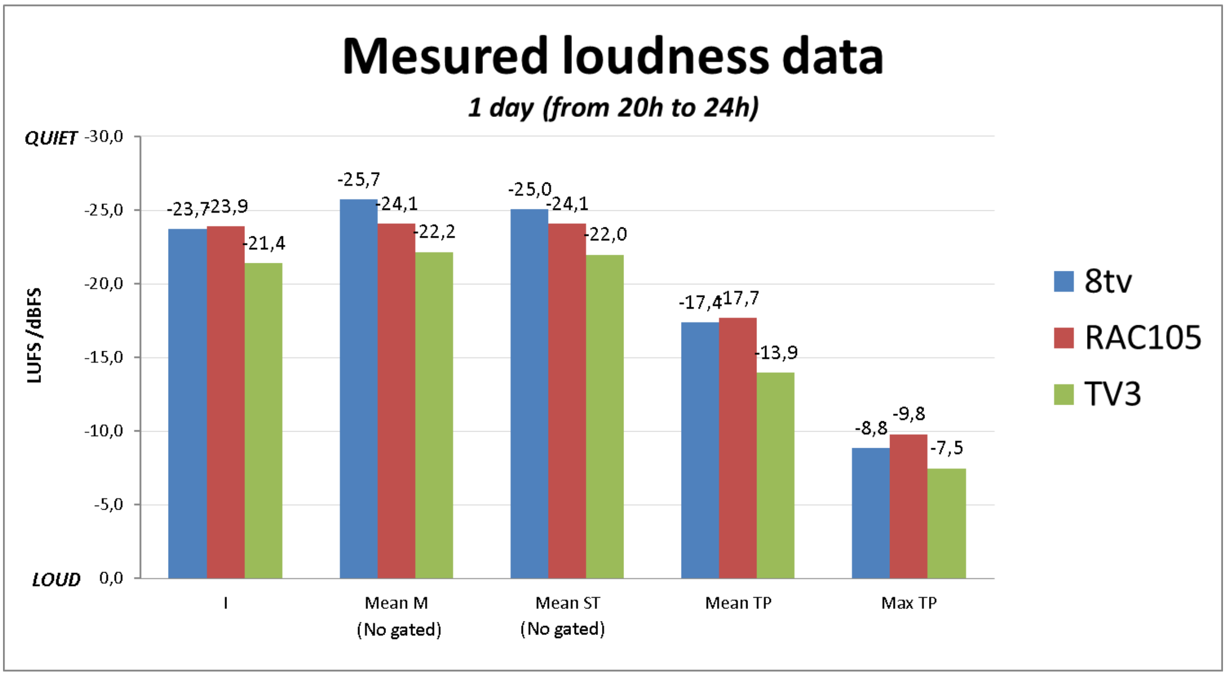


Figure 9-10: Comparative of loudness per channel from 20h to 24h

Comparing the values showed in the previous figure, the conclusions that we can extract are:

- The louder channel is clearly TV3 (in mean).
- In mean, we could say that 8tv and RAC105tv follow the ITU-R BS.1726 [11] (in brief, it says that the mean peak value has to be -18 dB FS and that the peaks does not have to exceed -9 dB FS, all measured with a QPPM).
- That TV3 does not follow R128 nor ITU-R BS.1726.

9.4.2 Per channel and type of content

Viewing the Figure 9-11 we can confirm that the audience is right when they complain saying that the commercials are louder than the rest of content (*).

(*) TV3 is not included in this comparison because we cannot reach to its asrun.

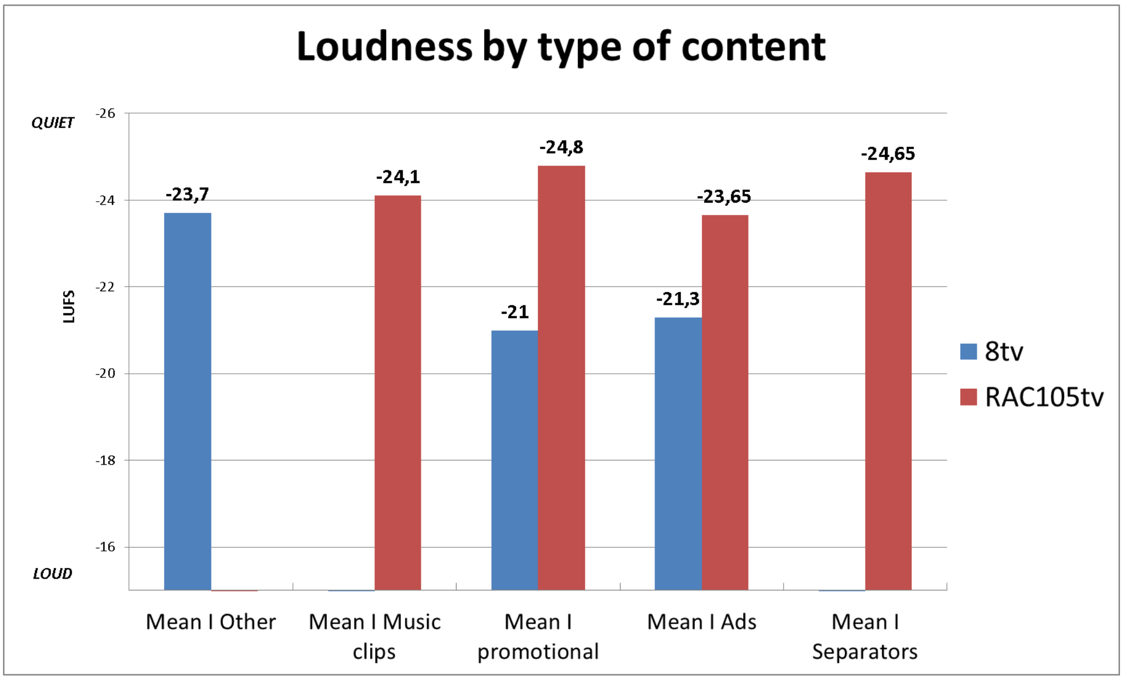


Figure 9-11: Comparative of loudness per channel and type of content

The conclusions that we can extract from the previous figure are:

- In mean, in 8tv the commercials and promotional content are more than 2 LU louder than other content.
- In RAC105tv, in mean, the louder content are the commercials as well, but the difference to music clips (the principal content in this channel) are only of 0.45 LU.

9.4.3 Histogram and LRA

As we can read in EBU Tech 3342 [35] the LRA value is directly related with loudness histogram and in the following figure we can see the ST loudness histogram of the captured data.

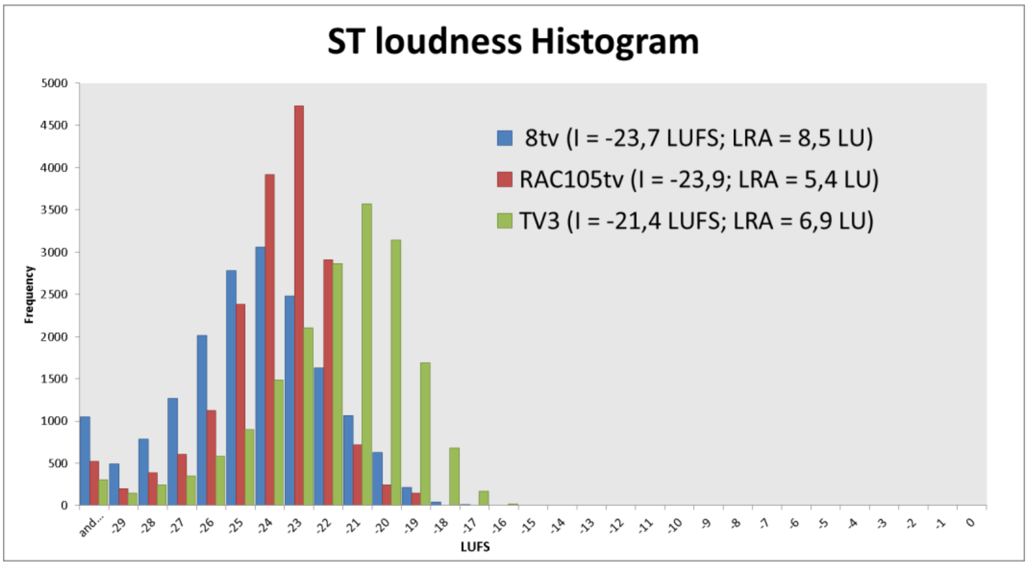


Figure 9-12: ST Histogram

Viewing the Figure 9-12 we can verify that the thinner histogram (most compressed audio signal) is the RAC105tv music channel, and it corresponds with the smallest LRA value.
The three LRA values are low, but remember that in TV the useful dynamic range is lower than, for instance, in the cinemas (see 3.2.5).
And we can see that the central value of the histograms it corresponds, obviously, to the I loudness value.

SECTION 3:
Final review

10 Future work

In broadcast technology the future work becomes faster in a present work, and loudness subject is an example, in 6 years (2006-2012) 4 different versions of the ITU-R BS.1770 have been released.

I would continue this work about loudness doing the following tasks:

- Use the LMS developed for this paper to evaluate the accomplishment of loudness recommendation by broadcasters. Specifically evaluate the loudness changes between ads and programs.
- Improving the performance of LME True Peak (TP) meter.
- Adapting the TP meter to ITU-R BS.1770-3 [17] (now is implemented following the ITU-R BS.1770-2 [27]).
- Developing a visual loudness meter (visual layer for LME).
- Implementing an easy to use plugin compatible with the most popular video / audio broadcast editors, in order to automatically correct loudness in tapeless content. For example Audio Units plug-ins (AU) [46], or Virtual Studio Technology plug-ins (VST) [47].
- Designing easy methods to apply loudness detection and correction to the broadcast chain for tapeless and live content.

11 Reviewing objectives

In this chapter we review the proposed objectives (see chapter 1) and we analyze if they have been passed or not.

We consider that we have reached the main objective proposed for this work that was "**explore the normalization of audio levels in the media industry**". We have studied the loudness motivation, the loudness background, the history, old and current standards, and we have designed, implemented, and tested in a broadcast environment a loudness meter that accomplish the last broadcast standards.

If we check the specific objectives:

- **Study and summarize background, references, and state of the art (competing) systems.**
    - We dedicated the chapter 3 to background and references, in the chapter 4, and 5 we have revised other kind of audio level meters and the current loudness standards. And finally in chapter 6 we have reviewed the state of the art of different kind of loudness meters.

- **Implement a Loudness meter following the last published standards.**
    - In chapter 7 we explained how we designed and implemented our loudness meter following the current broadcast standards. And we announced that it passed with high accuracy the standardization tests.

- **Evaluate the performance of the implemented loudness meter.**
    - In chapter 7.1.1 we evaluated the performance of our loudness meter and we obtained excellent results.

- **Design and implement a prototype of a complete loudness monitoring system based in our loudness meter.**
    - In chapter 7 we explain how we designed and implemented our loudness monitoring system (LMS) based in the implemented loudness meter. We also listed the most important design and implementation issues that we have found. And finally we measured three different broadcasters in order to check our LMS system (see chapter 9).

12 Conclusions

It has been a very interesting work directly related with my professional life. It was very useful for me because it allowed me to dig deeper into many audio concepts that I was working with for many years but I never thought about their deep meaning and history.

The main conclusions that I extracted from this master thesis are:

- I discovered that the concept of loudness that now is on cutting edge of broadcasting is a very old concept. I realized that the scientist community is studying how to measure the sound perception level for many years, even before radio and TV exists.

- I have seen that the old audio level meters (the majority becoming from the analog world) are inaccurate to measure the audio loudness. I understood the need for a new audio loudness measuring standard.

- I have realized that the loudness measuring technology is a new concept in broadcast technology that changes the way of work of the sound engineers, and it will deploy faster the following years.

- Nowadays you can find some loudness meters in the broadcast market: hardware, software, and integrated in other devices such mixers. **But noone can precisely link the loudness measurements with an asrun log allowing a great variety of analysis. This is my small contribution to broadcast loudness metering.**

- When you have to design and implement any complex system is not necessary to reinvent the wheel. Spend the time that you need analyzing the problem, look in the state of the art, divide the problem in separate layers, define clear and simple interfaces to communicate data between layers, and use the appropriate technology / programming language to implement each layer.

- By measuring loudness of different Spanish broadcasters I can certify the audience complains about louder ads in TV, and about different audio mean levels between channels.

- Finally I have to congratulate to EBU for R128. It is a well explained recommendation that greatly helps the people that want to work with it, all papers are free in the website, there are test files, videos, etc…

Finally I want to say that with this work I can certify a sentence that I heard from somebody years ago: **"you listen it and you will hear it, you study it and you will know it, you implement it and you will understand it".**

13 Acronyms table

EBU	European Broadcasting Union
ITU	International Telecommunications Union
ATSC	Advanced Television Systems Committee
CAC	Consell del Audiovisual de Catalunya
ANSI	American National Standards Institute
UPC	Universitat Politècnica de Catalunya
SPL	Sound Pressure Level
RMS	Root Mean Square
CALM	Commercial Advertisements Mitigation Act
RMS	Root Mean Square
AL	Alignment Level
ML	Measure Level
PML	Permitted Maximum Level
PPM	Peak programme meter
QPPM	Quasi peak programme meter
EU	European Union
ADC	Analog Digital Converter
DAC	Digital Analog Converter
DRC	Dynamic Range Control
LUFS	Loudness Unit Full Scale
LKFS	Loudness K-weighted Full Scale
LRA	Loudness range
LU	Loudness units
TP	True peak
maxTP	Maximun True Peak
DTT	Digital Terrestrial Television
RP	Recommended practice
ARIB	Association of Radio Industries and Business
MS	Microsoft
NTP	Network Time Protocol
TCP	Transport Control Protocol
GUI	Graphical User Interface
BDA	Broadcast Driver Architecture

14 References

[1] Goverment, USA, “Goverment Printing Office - CALM Law,” 16 12 2009. [Online]. Available: http://www.gpo.gov/fdsys/pkg/BILLS-111hr1084rfs/pdf/BILLS-111hr1084rfs.pdf. [Accessed 01 11 2012].

[2] EBU, “EBU R128: Loudness normalization and permitted maximun level of audio signals,” 01 08 2011. [Online]. Available: http://tech.ebu.ch/docs/r/r128.pdf. [Accessed 01 11 2012].

[3] B. Chittka L, “Wikipedia picture,” 28 04 2009. [Online]. Available: http://en.wikipedia.org/w/index.php?title=File:Anatomy_of_the_Human_Ear.svg&page=1. [Accessed 28 10 2012].

[4] A. Kliczek, “Wikipedia picture,” 11 03 2010. [Online]. Available: http://en.wikipedia.org/w/index.php?title=File:Sound_pressure_diagram.svg&page=1. [Accessed 28 10 2012].

[5] A. N. S. Institute, “American national psychoacoustical terminology S3.20,” ANSI, New York, 1973.

[6] H. Fletcher and W. Munson, “Loudness, its definition, measurement and calculation,” *Journal of the Acoustic Society of America,* 1933.

[7] ISO, “Acoustics - Normal equal-loudness-level contours,” ISO, 2003.

[8] P. J. Skirrow, “Wikipedia picture,” 14 09 2011. [Online]. Available: http://en.wikipedia.org/w/index.php?title=File:Lindos4.svg&page=1. [Accessed 28 10 2012].

[9] A. AB, “Advertising toolbox,” 2007. [Online]. Available: https://ecexpress.adtoox.com/ec/htmlloader.jsp?method=FileTechnicalSpecification&show=specification&group=SD. [Accessed 01 11 2012].

[10] ITU, “R BS.645-2,” 1992. [Online]. Available: http://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.645-2-199203-I!!PDF-E.pdf. [Accessed 03 11 2012].

[11] ITU, “R BS.1726,” 04 2005. [Online]. Available: http://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1726-0-200504-I!!PDF-E.pdf. [Accessed 04 11 2012].

[12] “Wikipedia,” 07 2007. [Online]. Available: http://en.wikipedia.org/wiki/Peak_programme_meter. [Accessed 26 12 2012].

[13] EBU Techincal, “EBU Tech 3205,” 11 1979. [Online]. Available: http://tech.ebu.ch/docs/tech/tech3205.pdf. [Accessed 03 11 2012].

[14] IEC, “IEC 60268-18: Sound system equipment - Peak programme level meters, Digital audio peak level meter,” IEC, Geneva, 1995.

[15] IEC, “IEC 60268-17: Sound system equipment - Standard volume indicators,” IEC, Geneva, 1990.

[16] ITU, “R-BS.1770: Algorithms to measure audio programme loudness and true-peak audio level,” 2006. [Online]. Available: http://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1770-0-200607-S!!PDF-E.pdf. [Accessed 17 11 2012].

[17] ITU, “R BS.1770-3: Algorithms to measure audio programme loudness and true-peak audio level,” 08 2012. [Online]. Available: http://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1770-3-201208-I!!PDF-E.pdf. [Accessed 17 11 2012].

[18] TC Electronic, “TC electronic,” 08 2012. [Online]. Available: http://www.tcelectronic.com/lm2-gallery.asp. [Accessed 18 11 2012].

[19] ISO, “ISO 532-1975: Method for calculating loudness level,” ISO, 1975.

[20] E. B. Brixen, Audio Metering, Denmark: Broadcast Publishing, 2001.

[21] IEC, “IEC 61672 - Sound level meters specifications,” IEC, Geneva, 2002.

[22] ITU, “ITU-R BS.468-4: Measurement of audio-frequency noise voltage level in sound broadcasting,” 1986. [Online]. Available: http://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.468-4-198607-I!!PDF-F.pdf. [Accessed 20 01 2013].

[23] “Wikipedia picture,” 11 12 2011. [Online]. Available: http://upload.wikimedia.org/wikipedia/commons/3/39/Acoustic_weighting_curves_%281%29.svg. [Accessed 22 12 2012].

[24] “Wikipedia picture,” 19 04 2009. [Online]. Available: http://upload.wikimedia.org/wikipedia/en/c/c2/Lindos3.svg. [Accessed 22 12 2012].

[25] G. A. Soulodre, *Evaluation of Objective Loudness Meters,* Audio Engineering Society, 2004.

[26] ITU, “R BS.1770-1: Algorithms to measure audio programme loudness and true-peak audio level,” 09 2007. [Online]. Available: http://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1770-1-200709-S!!PDF-E.pdf. [Accessed 17 11 2012].

[27] ITU, “R BS-1770-2: Algorithms to measure audio programme loudness and true-peak audio level,” 03 2011. [Online]. Available: http://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1770-2-201103-S!!PDF-E.pdf. [Accessed 17 11 2012].

[28] ATSC, “Digital audio compression standard (AC-3) rev. B,” 06 2005. [Online]. Available: http://www.atsc.org/cms/index.php/standards/document-download/doc_download/13-a52b-digital-audio-compression-standard-ac-3-e-ac-3-revision-b. [Accessed 05 01 2013].

[29] M. Babbitt, “Transmitting an Accurate Dialnorm Value Without Reinventing the Wheel,” 2005.

[30] Blue cat audio, “Blue Cat's FreqAnalyst,” [Online]. Available: http://www.bluecataudio.com//show_screenshot.php?product=Product_FreqAnalyst&image=main.png. [Accessed 11 01 2013].

[31] DK-Technologies, “DK-Technologies,” 01 2009. [Online]. Available: http://www.dk-technologies.com/downloads/specs/MSD100C-Loudness.pdf. [Accessed 18 11 2012].

[32] DK Technologies, “DK Technologies,” 2007. [Online]. Available: http://www.dk-technologies.com/downloads/StarFish.pdf. [Accessed 18 11 2012].

[33] ATSC, “A/85: Techniques for Establishing and Maintaining Loudness for Digital Television,” 09 11 2009. [Online]. Available: http://www.atsc.org/cms/index.php/standards/document-download/doc_download/33-a85-techniques-for-establishing-and-maintaining-audio-loudness-for-digital-television. [Accessed 01 01 2013].

[34] EBU, “Technical document 3341,” 08 2011. [Online]. Available: http://tech.ebu.ch/docs/tech/tech3341.pdf. [Accessed 05 01 2013].

[35] EBU, “Technical document 3342,” 08 2011. [Online]. Available: http://tech.ebu.ch/docs/tech/tech3342.pdf. [Accessed 05 01 2013].

[36] EBU, “Technical document 3343,” 08 2011. [Online]. Available: http://tech.ebu.ch/docs/tech/tech3343.pdf. [Accessed 05 01 2013].

[37] EBU, “Technical document 3344,” 10 2011. [Online]. Available: http://tech.ebu.ch/docs/tech/tech3344.pdf. [Accessed 05 01 2013].

[38] ATSC, “A/53 part 1 - Digital Television system,” 08 2009. [Online]. Available: http://www.atsc.org/cms/standards/a53/a_53-Part-1-2009.pdf. [Accessed 06 01 2013].

[39] ATSC, “A/53 part 5 - AC-3 Audio System Characteristics,” 07 2010. [Online]. Available: http://www.atsc.org/cms/standards/a53/a_53-Part-5-2010.pdf. [Accessed 07 01 2013].

[40] ATSC, “A/53 part 6 - Enhanced AC-3 Audio System Characteristics,” 06 07 2010. [Online]. Available: http://www.atsc.org/cms/standards/a53/a_53-Part-6-2010.pdf.

[Accessed 07 01 2013].

[41] EBU, “List of implementers supporting EBU R 128,” [Online]. Available: http://tech.ebu.ch/Jahia/site/tech/cache/offonce/loudness/loudness-faq-general. [Accessed 09 02 2013].

[42] Microsoft, “Directshow SDK,” [Online]. Available: http://msdn.microsoft.com/es-es/library/windows/desktop/dd375454. [Accessed 02 03 2013].

[43] GStreamer, “Gstreamer,” [Online]. Available: http://gstreamer.freedesktop.org/. [Accessed 02 03 2013].

[44] Apple, “Quicktime for developers,” [Online]. Available: https://developer.apple.com/quicktime/. [Accessed 02 03 2013].

[45] FFMpeg, “FFMpeg,” [Online]. Available: http://www.ffmpeg.org/index.html. [Accessed 02 03 2013].

[46] Apple, “Audio units for developpers,” [Online]. Available: https://developer.apple.com/library/mac/#documentation/MusicAudio/Conceptual/AudioUnitProgrammingGuide/Introduction/Introduction.html. [Accessed 02 03 2013].

[47] Steinberg, “VST Plug-ins developper,” [Online]. Available: http://www.steinberg.net/en/company/developer.html. [Accessed 02 03 2013].

[48] Tektronix, “Tektronix TG8000 specs,” [Online]. Available: http://www.tek.com/datasheet/multiformat-video-generator-1. [Accessed 29 03 2013].

[49] DekTec, “DeckTec,” [Online]. Available: http://www.dektec.com/index.asp. [Accessed 29 03 2013].

[50] Microsoft, “Broadcast Driver Architecture Drivers,” [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/hardware/ff556576(v=vs.85).aspx. [Accessed 29 03 2013].

[51] Vecotr3, “Vectorbox,” [Online]. Available: http://vector3.es/vectorboxes/cast/vectorbox.htm. [Accessed 05 04 2013].

[52] VSN, “VSN Multicom,” [Online]. Available: http://www.vsn-tv.com/es/programa/overview/61/VSNMULTICOM.html. [Accessed 13 04 2013].

[53] Barlovento comunicación, “Analisis de audiencias marzo 2013,” [Online]. Available: http://www.barloventocomunicacion.es/images/publicaciones/NOTA%20MARZO%202013%20BARLOVENTO%20COMUNICACION%20%20AUDIENCIAS.pdf. [Accessed 06 04 2013].

15 Annexes

15.1 Manual of command line Loudness meter

15.1.1 Introduction

The JOCLoudness application is a windows console applications that loads a wave file (.wav) and give us the following data of the file:

- Integrated loudness
- Vector of short term loudness
- Vector of momentary loudness
- Vector and maximum true peak level
- Loudness range value

15.1.2 Input modifiers

The accepted modifiers of JOCLoudness.exe are:

Name	***Mandatory***	***Description***
/audio file	YES	Input audio file (check accepted wav formats below)
/tin	NO	Trim in point in seconds. It is the point where starts to analyze loudness parameters.
/tout	NO	Trim out point in seconds. It is the point where stops to analyze loudness parameters.
/config	NO	Filename of the input config file (check config parameters below)
/outlouddata	NO	Filename of the output file with loudness data parameters
/outaudiofile	NO	Filename of the output file with audio processed segment (check wav out format below)
/logsfile	NO	Filename of the output log file
/verbose	NO	Display extra information to the screen

In the following lines you can see an example of use of JOCLoudness.exe modifiers:

```
> JOCLoudness.exe "c:\audiotest\totest.wav" /tin:1.1 /tout:20
/outaudiofile:"c:\out\outpartialfile.wav" /config:configcustom.ini
/outlouddata:"c:\dataloud\loud.txt" /logsfile:"c:\logsloud.log"
```

Example of use of JOCLoudness.exe

The previous command does the following tasks:

- It computes the loudness parameters of the ***"c:\audiotest\totest.wav"*** from ***1.1 seconds to 20 seconds.***

- It saves the loudness parameters to ***"c:\dataloud\loud.txt"*** using the configuration parameters loaded from ***"configcustom.ini"*** file.

- It saves the wav file ***"c:\out\outpartialfile.wav"*** that contains the audio segment from 1.1 seconds to 20 seconds of the input file.

- Append data to the ***"c:\logsloud.log"*** that contains the log data. Or it can create the logs file if no previous log file exists.

15.1.3 Loudness configuration file

The input configuration file is a .ini file (text format). The values that you can set in the configuration file are explained in the following table:

Name	***Default value***	***Value range***	***Used in Directshow module***	***Description***
action	loudness	loudness,filter	No	Sets the action to perform
MapChanel	[L R C Ls Rs 0]		Yes	Sets where the channels are located in audio input file. The first position of the vector indicates the first audio channel in the input file, second position is second channel, etc... L = Left channel R = Right channel C = Center channel Ls = Left surround channel Rs = Right surround channel 0 = Nothing (not use this channel)
ReadBlockSizeInSamplesPerChannel	1024	1... 2147483647	No	Audio samples per channel read in each iteration of the application loop
		TRUE PEAK SECTION		
TPCalc	0	0,1	Yes	Indicates if the application calculates the true peak value. We recommend set TPCalc = 0, it's a TIME CONSUMING FEATURE!!! (need to improve the oversampler LPF filter implementation)
TPDecaytimeMS	1700	1... 60000	Yes	[True peak config] The time needed in milliseconds since the audio peak is detected to the true peak meter shows the detected value minus TPDecayValueDB
TPDecayValueDB	-20	-0.1 ... -99999	Yes	[True peak config] Value substracted (in dB) to detected peak after TPDecaytimeMS milliseconds
TPRefreshInvervalMS	250	1... 60000	Yes	[True peak config] The refresh interval of true peak metter in miliseconds
		PRESET SELECTION SECTION		
Preset	r128	r128, a85, custom	Yes	Set the following configuration values according to R128 or A/85.

The following values only have effect if Preset = custom				
PRE FILTER SECTION (*)				
PreFilterCoefsAuto	1	0,1	Yes	Indicates if the application calculates the pre filter coefs automatically according to the sampling frequency of the input file. (*)
PreFilterACoefs	[1.0 -1.69065929318241 0.73248077421585]		Yes	Set the A coefs of the pre filter (Order 2 IIR filter) The coefs order is: [a0 a1 a2]
PreFilterBCoefs	[1.53512485958697 -2.69169618940638 1.19839281085285]		Yes	Set the B coefs of the pre filter (Order 2 IIR filter) The coefs order is: [b0 b1 b2]
RLB FILTER SECTION(*)				
RLBPreFilterCoefsAuto	1	0,1	Yes	Indicates if the application calculates the rlb filter coefs automatically according to the sampling frequency of the input file. (*)
RLBFilterACoefs	[1.0 -1.99004745483398 0.99007225036621]		Yes	Set the A coefs of the rlb filter (Order 2 IIR filter) The coefs order is: [a0 a1 a2]
RLBFilterBCoefs	[1.0 -2.0 1.0]		Yes	Set the B coefs of the pre filter (Order 2 IIR filter) The coefs order is: [b0 b1 b2]
MOMENTARY LOUDNESS PARAMETERS SECTION				
MomentaryAudioBlockDurationMS	400	1...3000	Yes	Sets the momentary audio block window duration in milliseconds
MomentaryAudioBlockOverlapping	0.75	0...1	Yes	Sets the windows overlapping of the momentary audio window
SHORT TERM LOUDNESS PARAMETERS SECTION				
ShortTermAudioBlockDurationMS	3000	1...10000	No	Sets the short term audio block window duration in milliseconds
ShortTermAudioBlockOverlapping	0.75	0...1	No	Sets the windows overlapping of the short term audio window
INTEGRATE LOUDNESS PARAMETERS SECTION				
IntegrateAlg	BS17702	BS17702, BS1770	No	Indicates which integrate algorithm will use the application
IntegrateGating	1	0,1	No	If the IntegrateAlg = BS17702, this parameter indicates if the integrate algorithm uses gating block or not
IntegrateAbsoluteThresholdDB	-70	0...-999999	No	If the IntegrateGating = 1, this parameter indicates the absolute threshold in LUFS

IntegrateRelativeThresholdDB	-10	-1...-999999	No	If the IntegrateGating = 1, this parameter indicates in LU the relative threshold applied to calculate the loudness integrate value
RESULTS PRESENTATION SECTION				
ResultPrecission	1	-1...20	No	Indicates the number of decimal positions in the results value -1 = max precission
ResultUnits	LUFS	LUFS, LKFS	No	Indicate the loudness units that shows the results file
LOUDNESS RANGE SECTION				
LRACalc	1	0,1	No	Indicates if the application calculates the loudness range values
LRAAudioBlockDurationMS	3000	1...10000	No	Sets the LRA audio block window duration in milliseconds
LRAAudioBlockOverlapping	0.75	0...1	No	Sets the windows overlapping of the LRA audio window
LRAAbsoluteThresholdDB	-70	0...-999999	No	Indicates the absolute threshold in LUFS used to calc LRA value
LRARelativeThresholdDB	-20	-1...-999999	No	Indicates in LU the relative threshold applied to calculate the LRA value
LRALowPercentile	10	0...100	No	Set the low percentile to calculate de LRA value
LRAHighPercentile	95	0...100	No	Set the high percentile to calculate de LRA value

(*) The ITU-R BS1770-2 and ITU-R BS1770 indicate the PRE and RLB filter coefficients referred to a sampling frequency of 48KHz. If the input audio file has a different audio sampling frequency and the coefficients are set in auto mode the application calculates the filter coefficients in order to keeps the filter weighting curve characteristics.

If no configuration file is set JOCLoudness.exe application will use the announced default configuration values.

15.1.4 Accepted input wave (.wav) formats

In the following table are showed the JOCLoudness.exe accepted wave (PCM) formats:

Header type	***Sample type***	***Sample freq.***	***Int sample bits***	***Float sample bits***
16 bytes 18 bytes 40 bytes (WAVEX)	Int float	All	8 16 24	32

15.1.5 Output wave format

The output wave format of JOCLoudness.exe is:

Header type	*Sample type*	*Sample freq.*	*Float sample bits*
16 bytes	float	All (same as input)	32

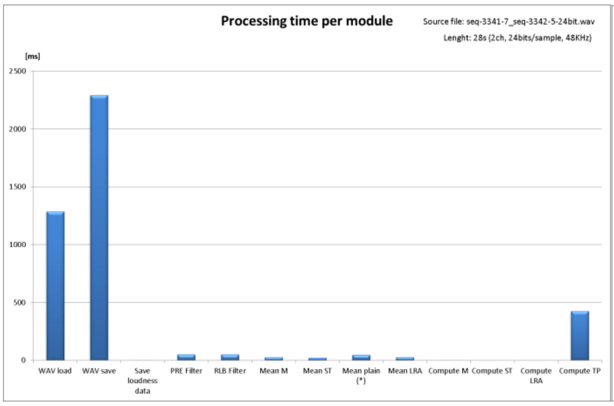
15.2 Measuring engine performance raw data

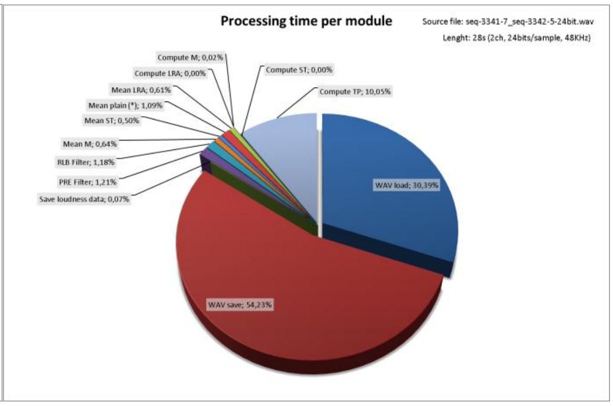
Performance analysys of loudness engine (2ch source)

Source file	seq-3341-7_seq-3342-5-24bit.wav
Source lenght	28000
Source channels	2
File format	PCM, 24b/sample, 48000KHz

Module description	*Short name*	*ALL ON RELEASE*	*% of total*	*Only I, M, ST RELEASE*	*% of total*
Wav load	WAV load	1285	30,39%	1242	86,79%
Wav save	WAV save	2293	54,23%	0	0,00%
Save loudness data to disk (calc I and LRA)	Save loudness data	3	0,07%	4	0,28%
PRE filter	PRE Filter	51	1,21%	39	2,73%
RLB filter	RLB Filter	50	1,18%	46	3,21%
Mean M loudness	Mean M	27	0,64%	30	2,10%
Mean ST loudness	Mean ST	21	0,50%	24	1,68%
Mean plain loudness (*)	Mean plain (*)	46	1,09%	42	2,94%
Mean LRA loudness	Mean LRA	26	0,61%	0	0,00%
Compute M (sum channels and calc log value)	Compute M	1	0,02%	4	0,28%
Compute ST (sum channels and calc log value)	Compute ST	0	0,00%	0	0,00%
Compute LRA data	Compute LRA	0	0,00%	0	0,00%
Compute true peak data	Compute TP	425	10,05%	0	0,00%
TOTAL:		4228	1,00	1431	1,00
Estimated time per channel:		2114		715,5	
			Times faster		Times faster
Real time:		0,151	6,622517	0,051107143	19,56673655
Estimated Real time per channel:		0,0755	13,24503	0,025553571	39,1334731
Estimated Real time per channel (without WAV read):				0,003375	296,2963

(*) Only used in NO gated measure



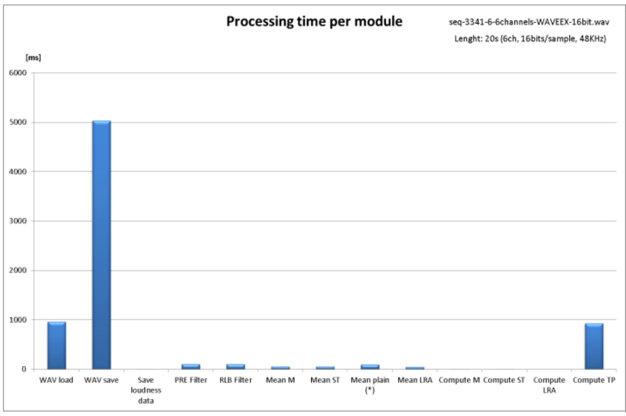


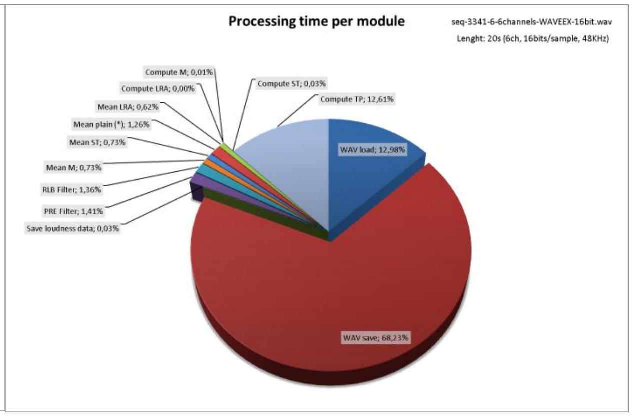
Performance analysys of loudness engine (6ch source)

Source file	seq-3341-6-6channels-WAVEEX-16bit.wav
Source lenght	20000
Source channels	6
File format	PCM, 16b/sample, 48000KHz

Module description	*Short name*	*ALL ON RELEASE*	*% of total*	*Only I, M, RELEASE*	*% of total*
Wav load:	WAV load	958	12,98%	933	68,91%
Wav save:	WAV save	5034	68,23%	0	0,00%
Save loudness data to disk (calc I and LRA)	Save loudness data	2	0,03%	6	0,44%
PRE filter:	PRE Filter	104	1,41%	110	8,12%
RLB filter:	RLB Filter	100	1,36%	99	7,31%
Mean M loudness:	Mean M	54	0,73%	53	3,91%
Mean ST loudness:	Mean ST	54	0,73%	59	4,36%
Mean plain loudness (*):	Mean plain (*)	93	1,26%	88	6,50%
Mean LRA loudness:	Mean LRA	46	0,62%	0	0,00%
Compute M (sum channels and calc log value)	Compute M	1	0,01%	5	0,37%
Compute ST (sum channels and calc log value)	Compute ST	2	0,03%	1	0,07%
Compute LRA data:	Compute LRA	0	0,00%	0	0,00%
Compute true peak data:	Compute TP	930	12,61%	0	0,00%
TOTAL TIME:		7378	1,00	1354	1,00
Estimated time per channel:		1229,667		225,6667	
			Times faster		Times faster
Real time:		0,3689	2,710762	0,0677	14,77104874
Estimated Real time per channel:		0,061483	16,26457	0,011283	88,62629247
Estimated Real time per channel (without WAV read):				0,003508	285,0356295

(*) Only used in NO gated measure





15.3 “EBU Mode” compliance test results

“EBU mode” minimum requirements for **loudness** measure (EBU Tech 3341):

Test case	*File*	*EBU Expected [LUFS]*	*EBU accepted tolerance*	*Our measuring engine result [LUFS]*
1	seq-3341-5-16bit-v02.wav	I = -23.0	+-0.1	**I = -23.0**
2	seq-3341-6-5channels-16bit.wav	I = -33.0	+-0.1	**I = -33.0**
3	seq-3341-3-16bit-v02.wav	I = -23.0	+-0.1	**I = -23.0**
4	seq-3341-4-16bit-v02.wav	I = -23.0	+-0.1	**I = -23.0**
5	seq-3341-5-16bit-v02.wav	I = -23.0	+-0.1	**I = -23.0**
6a	seq-3341-6-5channels-16bit.wav	I = -23.0	+-0.1	**I = -23.0**
6b	seq-3341-6-6channels-WAVEEX-16bit.wav	I = -23.0	+-0.1	**I = -23.0**
7	seq-3341-7_seq-3342-5-24bit.wav	I = -23.0	+-0.1	**I = -23.0**
8	seq-3341-2011-8_seq-3342-6-24bit-v02.wav	I = -23.0	+-0.1	**I = -23.0**

“EBU mode” minimum requirements for **loudness range** (LRA) measure (EBU Tech 3342):

Test case	*File*	*EBU Expected LRA [LU]*	*EBU accepted tolerance*	*Our measuring engine result [LU]*
1	seq-3342-1-16bit.wav	I = 10.0	+-1	**I = 10.0**
2	seq-3342-2-16bit.wav	I = 5.0	+-1	**I = 5.0**
3	seq-3342-3-16bit.wav	I = 20.0	+-1	**I = 20.0**
4	seq-3342-4-16bit.wav	I = 15.0	+-1	**I = 15.0**
5	seq-3341-7_seq-3342-5-24bit.wav	I = 5.0	+-1	**I = 4.9**
6	seq-3341-2011-8_seq-3342-6-24bit-v02.wav	I = 15.0	+-1	**I = 14.9**

15.4 Manual of CJOCLoudnessDS directshow filter

15.4.1 Introduction

The JOCLoudnessDS is a directshow filter that calculates the following loudness values: Momentary Loudness, Short Term Loudness, and true peak.

The use of directshow technology allows us to use any file or device as audio source for Loudness meter; we don't have to worry about codecs or device drivers. If we have installed the correct codecs or the device has a directshow driver we can calculate the loudness values of the audio source.

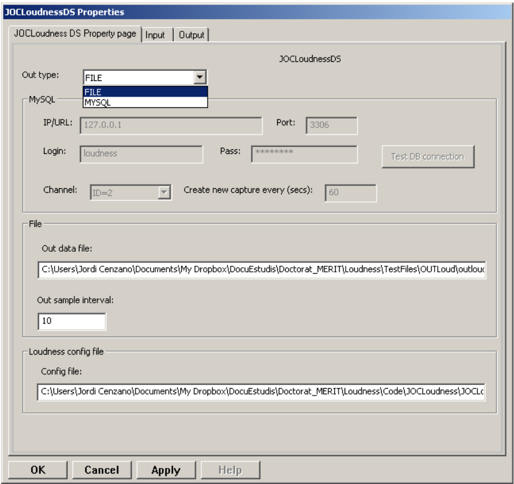
15.4.2 Input audio data

Could be any of follow:

- Video or audio file that directshow can decode.
- Digital Terrestrial Television (DTT) live signal decoded using a Broadcast driver Architecture (BDA) hardware compliant, for instance: PCTV DTT receivers or AverMedia USB DTT receivers, etc..
- Any video/audio captured live signal (analog, SDI, AES, etc...). You can use any device that has a compliant driver in directshow architecture, for instance: Blackmagic Decklink capture cards, Osprey capture cards, etc…

15.4.3 Filter configuration parameters

The configuration values are accessible using JOCLoudnessDS property page or querying the IID_IJOCLoudnessDS interface.



JOCLoudnessDS filter properties page

```
// {DC9D21D9-B96C-419E-8D4B-B0FACC5E6A63}
static const GUID CLSID_JOCLoudnessDS =
{ 0xdc9d21d9, 0xb96c, 0x419e, { 0x8d, 0x4b, 0xb0, 0xfa, 0xcc, 0x5e, 0x6a, 0x63 } };

// {872889F1-4947-4125-8DCC-0810E82E3BC8}
static const GUID IID_IJOCLoudnessDS =
{ 0x872889f1, 0x4947, 0x4125, { 0x8d, 0xcc, 0x8, 0x10, 0xe8, 0x2e, 0x3b, 0xc8 } };

// {47C1E123-BE81-4658-87DE-EB979060CC61}
static const GUID CLSID_jocLoudnessDS_PropPage =
{ 0x47c1e123, 0xbe81, 0x4658, { 0x87, 0xde, 0xeb, 0x97, 0x90, 0x60, 0xcc, 0x61 } };
```

JOCLoudnessDS CLSID definitions

The available configurations parameters are:

Name	***Description***
Out type	The desired output mode ("FILE" or "MYSQL"): "FILE": It saves the loudness data to a file "MYSQL": It inserts the data into a mySQL database (see 15.5)
Loudness configuration file	Indicates the file where are the low level parameters used to configure our loudness meter engine. It is the same file used for the command line loudness meter, for more information see the command line loudness meter configuration file in chapter 15.1.3. JOCLoudnessDS only uses the following parameters of the configuration file: ***MapChanel*** ***TPDecaytimeMS*** ***TPDecayValueDB*** ***TPRefreshInvervalMS*** ***Preset*** ***PreFilterCoefsAuto*** ***PreFilterACoefs*** ***PreFilterBCoefs*** ***RLBPreFilterCoefsAuto*** ***RLBFilterACoefs*** ***RLBFilterBCoefs*** ***MomentaryAudioBlockDurationMS*** ***MomentaryAudioBlockOverlapping*** ***ShortTermAudioBlockDurationMS*** ***ShortTermAudioBlockOverlapping***

If the option “FILE” is selected the enabled configuration parameters will be:

FILE ***Name***	***Description***
Out data file	Sets the output loudness data file path. For instance: ***c:\loudnessdata.txt***
Out sample interval	This parameter indicates the number of loudness samples that will be saved into file in every file access. It is used to improve the file access efficiency

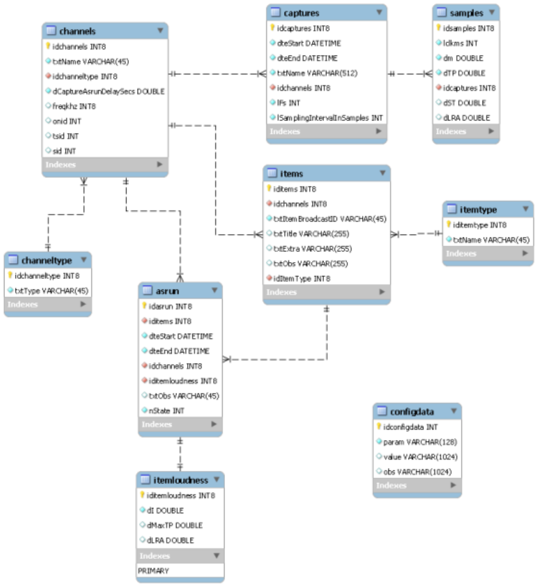
If the option “MYSQL” is selected the enabled configuration parameters will be:

MYSQL ***Name***	***Description***
IP/URL	Indicates the name or the IP of the computer where the MySQL server is installed.
Port	Indicates the port used to connect to MySQL server
Login	login used to connect to "lodnessdb" schema
Pass	Password used to connect to "lodnessdb" schema
Channel	Indicates in witch channel will be assigned the future captured samples. To refresh this value click in "Test DB connection" with a valid connection parameters.
Create new capture every	In order to minimize the clock drifting (between asrun clock and samples clock), a new capture is created every time interval. No samples are lost in this action. Default value: ***60sec***. (the units of this value are in seconds)

15.5 LMS Database description

15.5.1 Schema

If you select to send the loudness data to MySQL database (Out type = MYSQL). The database where the loudness data will send has to be the structure showed in the following figure:



Loudness DB schema

15.5.2 Tables

Here you can find a description of DB tables and columns.

15.5.2.1 Channels

This table stores the DTT channel data.

Column name	*Description*
Idchannels	Key
txtName	Channel name
Idchanneltype	[ext key] Indicates the channel type, See 15.5.2.2
dCaptureArunDelaySecs	Indicates the delay between asrun time and and DTT decoded time (*)
Freqkhz	The channel DTT frequency in KHz. Used by DTT module to tune the channel
ONID	Original network ID. A unique network identifier. Used by DTT module to tune the channel
TSID	The transport stream ID. Unique identifier of transport stream. Used by DTT module to tune the channel
SID	Service ID. Identifies the channel to tune inside a transport stream. Used by DTT module to tune the channel

(*) This value indicates the DTT broadcast chain delay, it is a fixed delay that usually are around 3 seconds.

15.5.2.2 ChannelType

This table stores the DTT channel data.

Column name	Description
Idchanneltype	Key
txtType	Indicates if the channel is a “TDT” (default) or “FILE”

15.5.2.3 Asrun

This table stores the broadcasted events imported from playout asrun files.

Column name	Description
Idasrun	Key
Iditems	[ext key] Reference broadcast item data, See 15.5.2.5
dteStart	Timestamp that indicates when this item started to broadcast
dteEnd	Timestamp that indicates when this item finished to broadcast
Idchannels	[ext key] Reference to the channel that belongs this broadcast event, See 15.5.2.1
idItemLoudness	[ext key] Reference to the loudness data of this broadcast event, See 15.5.2.4
txtObs	Extra text field used for description purposes
nState	Broadcast event state. Can be: 0 = Loudness computation pending 1 = Loudness computation in progress 2 = Loudness computation done 3 = Loudness computation error

15.5.2.4 itemLoudness

This table stores the loudness data of the broadcasted events.

Column name	Description
idItemLoudness	Key
dI	The loudness integrated (I) value of the broadcast event
dMaxTP	The maximum True Peak (maxTP) value of the broadcast event
dLRA	The loudness range (LRA) value of the broadcast event

15.5.2.5 Items

This table stores the broadcast items data.

Column name	Description
idItems	Key
Idchannels	[ext key] Reference to the channel that belongs this broadcast item, See 15.5.2.1
txtItemBroadcastID	Unique identifier of the broadcast item
txtTitle	Title of the broadcast item
txtExtra	Extra information of the broadcast item
txtObs	Extra text field used for description purposes
idItemType	[ext key] Indicates the broadcast item type. Reference to item type, See 15.5.2.6

15.5.2.6 ItemType

It indicates the type of content of broadcast items, can be ads, teasers, programme, etc…

Column name	Description
idItemType	Key
txtName	Name of the item type

15.5.2.7 Captures

It stores the data of the captured loudness group of samples.

Column name	Description
Idcaptures	Key
dteStart	Timestamp that indicates when started to capture the referenced group of loudness samples
dteEnd	Timestamp that indicates when finished to capture the referenced group of loudness samples
txtName	Name of this capture
idChannels	[ext key] Reference to the channel that belongs this capture, See 15.5.2.1
lFS	Frequency sampling of the source audio signal of this capture
lSamplingIntervalInsamples	The time gap (in audio samples) between every sample capture. In R128 is 100ms by default

15.5.2.8 Samples

This table stores the data of every loudness sample.

Column name	*Description*
Idsamples	Key
Lclkms	Sample timestamp (in ms) since capture started
dm	Loudness momentary value
dTP	Loudness instantaneous true peak (TP) value
Idcaptures	[ext key] Reference to the capture that belongs this sample entry, See 15.5.2.7
dST	Loudness short term (ST) value (0 = not computed) (*)
dLRA	Loudness value that will be used to compute the loudness range of the broadcast event/item (0 = not computed) (**)

(*)In R128, using an overlap factor of 75% the ST value is computed every 7.5 momentary values.

(**)The refresh rate of LRA will vary depending on *LRAAudioBlockDurationMS* and *LRAAudioBlockOverlapping* config values.
In our R128 preset the default values are: *LRAAudioBlockDurationMS* = 3s, and *LRAAudioBlockOverlapping* = 0.75. With these config parameters the dLRA value will be computed every 7.5 momentary values.

15.5.2.9 Configdata

This was created to store global system configuration parameters.

Column name	*Description*
Idconfigdata	Key
Param	Parameter name
Value	Parameter value
Obs	Extra text field used for description purposes

15.5.3 SQL creation code

You can create the database using the following SQL sentences (tested with mySQL):

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

DROP SCHEMA IF EXISTS `LoudnessDB` ;
CREATE SCHEMA IF NOT EXISTS `LoudnessDB` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci ;
USE `LoudnessDB` ;

-- -----------------------------------------------------
-- Table `LoudnessDB`.`channeltype`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `LoudnessDB`.`channeltype` ;

CREATE  TABLE IF NOT EXISTS `LoudnessDB`.`channeltype` (
  `idchanneltype` BIGINT(20) NOT NULL AUTO_INCREMENT ,
  `txtType` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`idchanneltype`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `LoudnessDB`.`channels`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `LoudnessDB`.`channels` ;

CREATE  TABLE IF NOT EXISTS `LoudnessDB`.`channels` (
  `idchannels` BIGINT(20) NOT NULL AUTO_INCREMENT ,
  `txtName` VARCHAR(45) NOT NULL ,
  `idchanneltype` BIGINT(20) NOT NULL ,
  `dCaptureAsrunDelaySecs` DOUBLE ZEROFILL NOT NULL ,
  `freqkhz` BIGINT(20) NULL ,
  `onid` INT NULL ,
  `tsid` INT NULL ,
  `sid` INT NULL ,
  PRIMARY KEY (`idchannels`) ,
  INDEX `fkchannels_idchanneltype` (`idchanneltype` ASC) ,
  CONSTRAINT `fkchannels_idchanneltype`
    FOREIGN KEY (`idchanneltype` )
    REFERENCES `LoudnessDB`.`channeltype` (`idchanneltype` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `LoudnessDB`.`captures`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `LoudnessDB`.`captures` ;

CREATE  TABLE IF NOT EXISTS `LoudnessDB`.`captures` (
  `idcaptures` BIGINT(20) NOT NULL AUTO_INCREMENT ,
  `dteStart` DATETIME NOT NULL ,
  `dteEnd` DATETIME NOT NULL ,
  `txtName` VARCHAR(512) NOT NULL ,
  `idchannels` BIGINT(20) NOT NULL ,
  `lFs` INT NOT NULL ,
  `lSamplingIntervalInSamples` INT NOT NULL ,
  PRIMARY KEY (`idcaptures`) ,
  INDEX `fkcaptures_idchannels` (`idchannels` ASC) ,
  CONSTRAINT `fkcaptures_idchannels`
    FOREIGN KEY (`idchannels` )
    REFERENCES `LoudnessDB`.`channels` (`idchannels` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `LoudnessDB`.`samples`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `LoudnessDB`.`samples` ;

CREATE  TABLE IF NOT EXISTS `LoudnessDB`.`samples` (
  `idsamples` BIGINT(20) NOT NULL AUTO_INCREMENT ,
```

```
  `lclkms` INT UNSIGNED NOT NULL ,
  `dm` DOUBLE NOT NULL ,
  `dTP` DOUBLE NOT NULL ,
  `idcaptures` BIGINT(20) NOT NULL ,
  `dST` DOUBLE NULL ,
  `dLRA` DOUBLE NULL ,
  PRIMARY KEY (`idsamples`) ,
  INDEX `fksamples_idcaptures` (`idcaptures` ASC) ,
  CONSTRAINT `fksamples_idcaptures`
    FOREIGN KEY (`idcaptures` )
    REFERENCES `LoudnessDB`.`captures` (`idcaptures` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `LoudnessDB`.`itemtype`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `LoudnessDB`.`itemtype` ;

CREATE  TABLE IF NOT EXISTS `LoudnessDB`.`itemtype` (
  `iditemtype` BIGINT(20) NOT NULL AUTO_INCREMENT ,
  `txtName` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`iditemtype`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `LoudnessDB`.`items`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `LoudnessDB`.`items` ;

CREATE  TABLE IF NOT EXISTS `LoudnessDB`.`items` (
  `iditems` BIGINT(20) NOT NULL AUTO_INCREMENT ,
  `idchannels` BIGINT(20) NOT NULL ,
  `txtItemBroadcastID` VARCHAR(45) NOT NULL ,
  `txtTitle` VARCHAR(255) NULL ,
  `txtExtra` VARCHAR(255) NULL ,
  `txtObs` VARCHAR(255) NULL ,
  `idItemType` BIGINT(20) NOT NULL ,
  PRIMARY KEY (`iditems`) ,
  INDEX `fkitems_itemtype` (`idItemType` ASC) ,
  INDEX `fkitems_channels` (`idchannels` ASC) ,
  CONSTRAINT `fkitems_itemtype`
    FOREIGN KEY (`idItemType` )
    REFERENCES `LoudnessDB`.`itemtype` (`iditemtype` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fkitems_channels`
    FOREIGN KEY (`idchannels` )
    REFERENCES `LoudnessDB`.`channels` (`idchannels` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `LoudnessDB`.`itemloudness`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `LoudnessDB`.`itemloudness` ;

CREATE  TABLE IF NOT EXISTS `LoudnessDB`.`itemloudness` (
  `iditemloudness` BIGINT(20) NOT NULL AUTO_INCREMENT ,
  `dI` DOUBLE ZEROFILL NOT NULL ,
  `dMaxTP` DOUBLE NULL ,
  `dLRA` DOUBLE NULL ,
  PRIMARY KEY (`iditemloudness`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `LoudnessDB`.`asrun`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `LoudnessDB`.`asrun` ;

CREATE  TABLE IF NOT EXISTS `LoudnessDB`.`asrun` (
  `idasrun` BIGINT(20) NOT NULL AUTO_INCREMENT ,
  `iditems` BIGINT(20) NOT NULL ,
  `dteStart` DATETIME NOT NULL ,
  `dteEnd` DATETIME NOT NULL ,
  `idchannels` BIGINT(20) NOT NULL ,
```

```sql
  `iditemloudness` BIGINT(20) NOT NULL ,
  `txtObs` VARCHAR(45) NULL ,
  `nState` INT ZEROFILL NOT NULL DEFAULT 0 ,
  PRIMARY KEY (`idasrun`) ,
  INDEX `fkasrun_iditems` (`iditems` ASC) ,
  INDEX `fkasrun_idchannels` (`idchannels` ASC) ,
  INDEX `fkasrun_iditemloudness` (`iditemloudness` ASC) ,
  CONSTRAINT `fkasrun_iditems`
    FOREIGN KEY (`iditems` )
    REFERENCES `LoudnessDB`.`items` (`iditems` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fkasrun_idchannels`
    FOREIGN KEY (`idchannels` )
    REFERENCES `LoudnessDB`.`channels` (`idchannels` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fkasrun_iditemloudness`
    FOREIGN KEY (`iditemloudness` )
    REFERENCES `LoudnessDB`.`itemloudness` (`iditemloudness` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `LoudnessDB`.`configdata`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `LoudnessDB`.`configdata` ;

CREATE  TABLE IF NOT EXISTS `LoudnessDB`.`configdata` (
  `idconfigdata` INT NOT NULL AUTO_INCREMENT ,
  `param` VARCHAR(128) NOT NULL ,
  `value` VARCHAR(1024) NULL ,
  `obs` VARCHAR(1024) NULL ,
  PRIMARY KEY (`idconfigdata`) )
ENGINE = InnoDB;



SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

-- -----------------------------------------------------
-- Data for table `LoudnessDB`.`channeltype`
-- -----------------------------------------------------
START TRANSACTION;
USE `LoudnessDB`;
INSERT INTO `LoudnessDB`.`channeltype` (`idchanneltype`, `txtType`) VALUES (1, 'FILE');
INSERT INTO `LoudnessDB`.`channeltype` (`idchanneltype`, `txtType`) VALUES (2, 'TDT');

COMMIT;

-- -----------------------------------------------------
-- Data for table `LoudnessDB`.`channels`
-- -----------------------------------------------------
START TRANSACTION;
USE `LoudnessDB`;
INSERT INTO `LoudnessDB`.`channels` (`idchannels`, `txtName`, `idchanneltype`, `dCaptureAsrunDelaySecs`,
`freqkhz`, `onid`, `tsid`, `sid`) VALUES (1, 'FROM FILE', 1, 0, NULL, NULL, NULL, NULL);
INSERT INTO `LoudnessDB`.`channels` (`idchannels`, `txtName`, `idchanneltype`, `dCaptureAsrunDelaySecs`,
`freqkhz`, `onid`, `tsid`, `sid`) VALUES (2, '8tv - BCN', 2, 3, 570000, 66, 1007, 10560);
INSERT INTO `LoudnessDB`.`channels` (`idchannels`, `txtName`, `idchanneltype`, `dCaptureAsrunDelaySecs`,
`freqkhz`, `onid`, `tsid`, `sid`) VALUES (3, 'RAC105tv - BCN', 2, 3, 570000, 66, 1007, 10563);

COMMIT;

-- -----------------------------------------------------
-- Data for table `LoudnessDB`.`itemtype`
-- -----------------------------------------------------
START TRANSACTION;
USE `LoudnessDB`;
INSERT INTO `LoudnessDB`.`itemtype` (`iditemtype`, `txtName`) VALUES (1, 'OTROS');
INSERT INTO `LoudnessDB`.`itemtype` (`iditemtype`, `txtName`) VALUES (2, 'PUBLI');
INSERT INTO `LoudnessDB`.`itemtype` (`iditemtype`, `txtName`) VALUES (3, 'SEP');
INSERT INTO `LoudnessDB`.`itemtype` (`iditemtype`, `txtName`) VALUES (4, 'PROMO');
INSERT INTO `LoudnessDB`.`itemtype` (`iditemtype`, `txtName`) VALUES (5, 'CLIP');

COMMIT;

-- -----------------------------------------------------
```

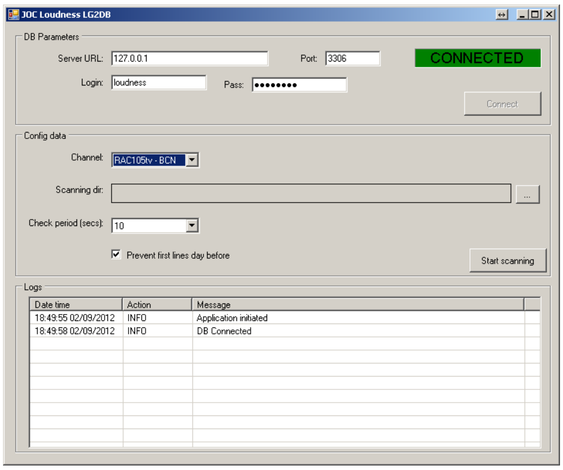
```sql
-- Data for table `LoudnessDB`.`configdata`
-- -----------------------------------------------------
START TRANSACTION;
USE `LoudnessDB`;
INSERT INTO `LoudnessDB`.`configdata` (`idconfigdata`, `param`, `value`, `obs`) VALUES (1, 'preset', 'r128',
'used to calc I, LRA, and TP');

COMMIT;
```

Loudness DB schema SQL creation code

15.6 Manual of LMS core modules

15.6.1 Manual of log to DB LMS core module

The function of this module is to scan the selected directory and when it detects a new asrun file it reads the file and it sends the broadcasted events data to DB.



GUI of log to DB module

Parameter Name	*Description*
Server URL	Indicates the IP or DNS of the computer where the central DB is installed
Port	Indicates the TCP port that this modules will use to establish the communication channel with central database
Login	The login that this module will use to access to DB
Pass	The password that this module will use to access to DB
Channel	The channel where the broadcasted events read from asrun file will be inserted
Scanning dir	The directory where this module will try to find the asrun files
Check period	The interval in seconds between two scanning
Prevent first line day before	An intelligent routine to handle with items that started to broadcast a day X and finished the day after (*)

(*) The algorithm “Prevent first line day before” does the following actions:

- It analyzes the start time of the firsts 10 events of asun log, and if the hours of the start time are greater or equal than 23 assigns that event to the day before.

- It analyzes the start time of the last 10 events of asun log, and if the hours of the start time are less than 01 assigns that event to the day after.

Finally to start scan you have to press “Start scanning” button.

15.6.2 Compliant asrun file syntax

The file asrun file has to be an ASCII text file.
The asrun file name has to accomplish the following structure:

Playoutlogs_*YYYYMMDD*.log

Where **YYYY** = Year, **MM** = Month, **DD** = day

In the following figure we can see a minimum sample of the file syntax:

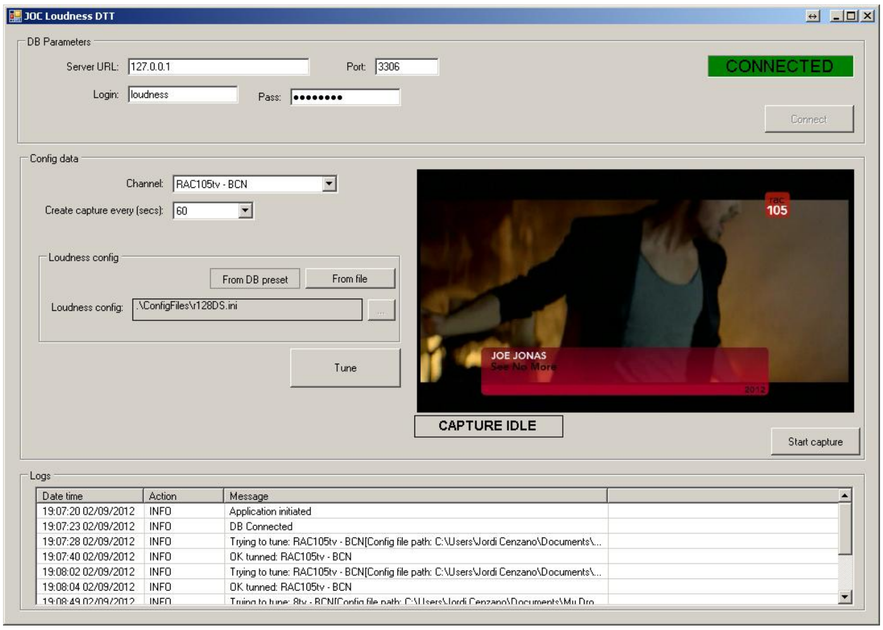
```
DISK    00:01:01        00:01:14        00:00:13:08     Ok      SEP_R105_169_0041
DISK    00:01:14        00:05:52        00:04:37:13     Ok      TV01635
DISK    00:05:52        00:09:20        00:03:28:06     Ok      TV02043
DISK    00:09:20        00:09:27        00:00:07:00     Ok      SEP_R105G_0006
DISK    00:09:27        00:14:38        00:05:10:10     Ok      TV00283
DISK    00:14:38        00:19:31        00:04:53:08     Ok      TV00367
...
```

Sample of the minimum asrun log file

Parameter order	***Description***
1	Have to be fixed to “DISK”
2	Start time. The time stamp when the clip stared to broadcast in HH:MM:SS
3	End time. The time stamp when the clip finished to broadcast in HH:MM:SS
4	The clip duration in HH:MM:SS:FF
5	The status of the broadcasted clip. Could be “Ok or “Error”
6	The clip unique ID

15.6.3 Manual of DTT LMS core module

This module tunes a DTT channel, computes the loudness parameters of its audio, and it sends the computed loudness data (Momentary, Short Term, and True Peak) to the central database.



GUI of DTT module

Name	*Description*
Server URL	Indicates the IP or DNS of the computer where the central DB is installed
Port	Indicates the TCP port that this modules will use to establish the communication channel with central database
Login	The login that this module will use to access to DB
Pass	The password that this module will use to access to DB
Channel	It shows the channels that you can tune. The tuning data are read from database, see 15.5.2.1
Create capture every	Indicates the time interval which the captures timestamp is refreshed. It is used to avoid the time drift problem described in 8.3.2
From DB preset	Indicates that the configuration parameters of LME will be set according to the “preset” field of configdata table (*), see 15.5.2.9
From file	You can select your personalized LME configuration file. The format of the file are explained in 15.4.3

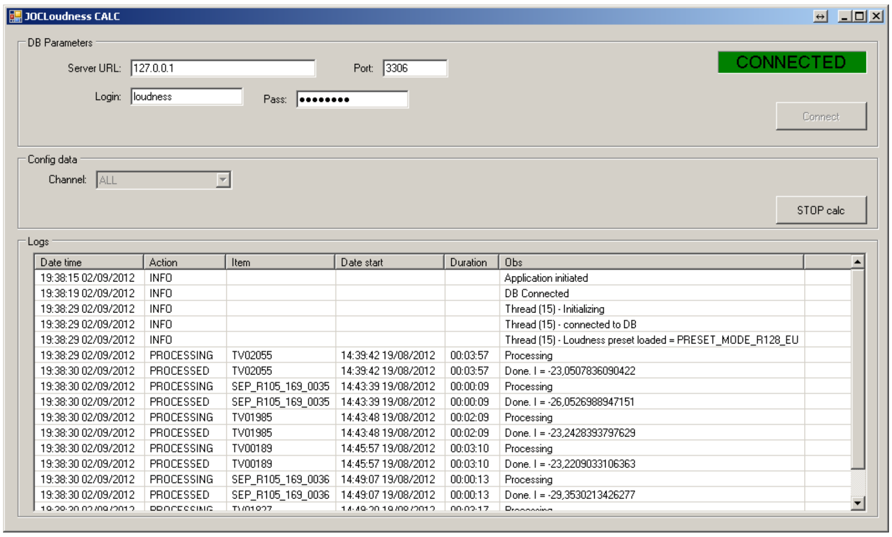
(*) The “preset” parameters can have these two values: “r128” and “a85”

To start DTT loudness analysis you only have to:

- Select the desired channel.
- Press “Tune”.
- Press “Start capture”.

15.6.4 Manual of CALC LMS core module

This module computes the loudness parameters of the broadcasted events using the captured loudness raw data.



GUI of CALC module

Name	***Description***
Server URL	Indicates the IP or DNS of the computer where the central DB is installed
Port	Indicates the TCP port that this modules will use to establish the communication channel with central database
Login	The login that this module will use to access to DB
Pass	The password that this module will use to access to DB
Channel	Indicates the channel that belong the items that you want to compute its loudness parameters

As you can see it is a very simple user interface. You only have to select the channel that you want to scan, or “ALL” if you want to that this instance scans all available channels, and finally press “START calc”.

15.6.5 Manual of SHOW LMS core module

The main feature of this module is to create and show reports using all data collected by the other LMS modules.

From the main form you can access to different kind of reports, in the following sections we will explain how to create all available reports, and how to analyze the loudness.

15.6.5.1 Report by broadcasted event

This report shows the loudness graph of a single broadcasted event.

Follow the instructions to create a loudness report of a broadcasted event:

<u>FORM A:</u>

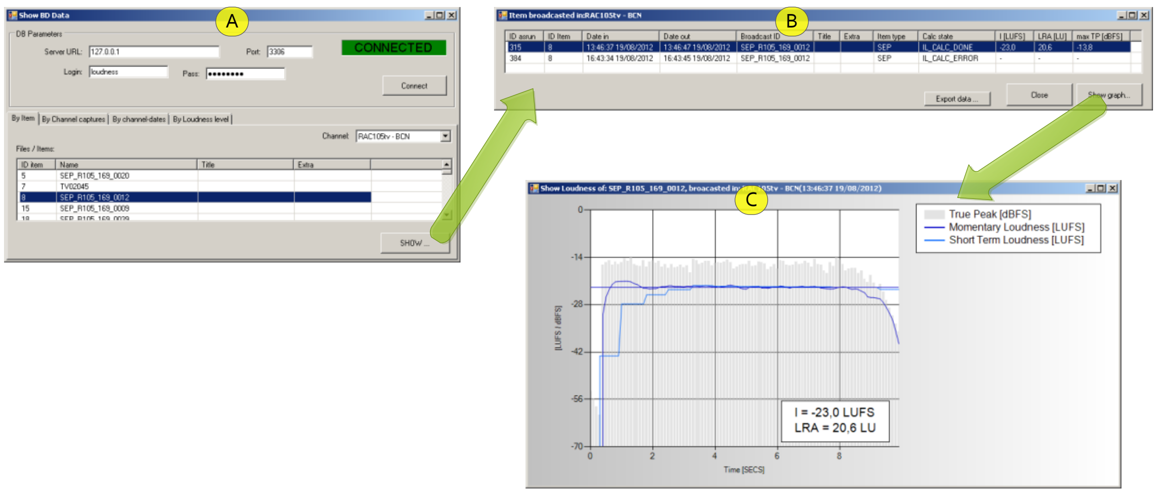
1. In the main tab section select “By Item”.
2. In the “Channel” combobox select the channel that belong the broadcasted item that you want to analyze.
3. In the items list select the broadcasted item which you want to create the report.
4. Press “SHOW…”, and the form B will appear.

<u>FORM B:</u>

1. Select the broadcasted event of the item that you want to see. You can sort the broadcasted events by different criteria by clicking on the headers of the list.
2. Press “Show graph…”.

<u>FORM C:</u>

1. In this form you can see and navigate in the loudness graph of the broadcasted event of the selected item. In the section 15.7 is explained how to read and navigate in loudness graphs.



SHOW module main form – Report by broadcasted item

Name	Description
Server URL	Indicates the IP or DNS of the computer where the central DB is installed
Port	Indicates the TCP port that this modules will use to establish the communication channel with central database
Login	The login that this module will use to access to DB
Pass	The password that this module will use to access to DB
Channel	Indicates the channel that belong the items that you want to create the loudness report
Files/Items	List of broadcasted items of the selected channel

15.6.5.2 Report by channel capture

This kind of report shows the loudness graph of the selected channel time span.

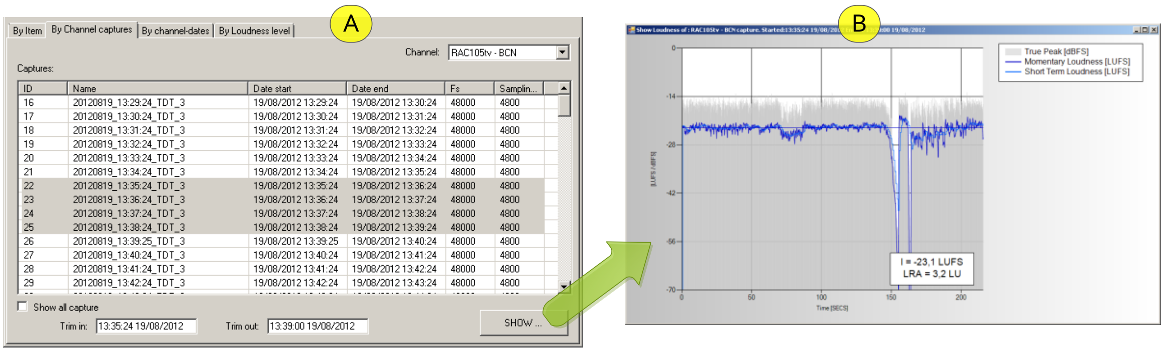
Follow the instructions to create a loudness report of a channel capture:

<u>FORM A:</u>

1. In the main tab section select “By channel captures”.
2. In the “Channel” combobox select the channel you want to analyze.
3. In the captures list you can select one capture or a group of them (pressing shift key).
4. If you want to trim the capture (or captures) you can uncheck “Show all capture” and edit the time in and time out of the captures.
5. Finally press “SHOW…”, and the form B will appear.

<u>FORM B:</u>

1. In this form you can see and navigate by a loudness graph of the selected channel time span. In the section 15.7 is explained how to read and navigate in loudness graphs.



SHOW module main form – Report by channel capture

Name	Description
Channel	Indicates the channel that you want to create the loudness report
Captures	List of captures of the selected channel (remember that 1 capture is created every 60s by default)
Show all capture	If it is checked all selected time span will be showed in the graph
Trim in	If “Show all capture” is unchecked indicates the start point of the loudness graph
Trim out	If “Show all capture” is unchecked indicates the end point of the loudness graph

15.6.5.3 Report by channel dates

This kind of report shows the loudness graph mixed with the asrun data of the selected channel time span.

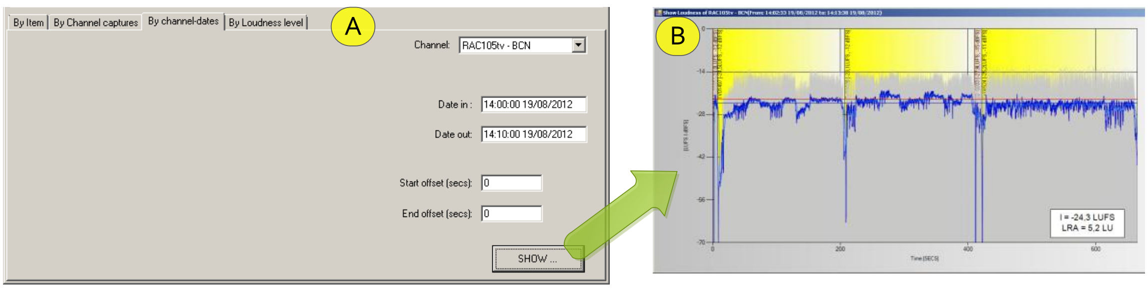
Follow the instructions to create a loudness report of a channel date span.

FORM A:

1. In the main tab section select “By channel-dates”.
2. In the “Channel” combobox select the channel you want to analyze.
3. Enter the start time of loudness analysis in “Date in”.
4. Enter the end time of loudness analysis in “Date out”.
5. Finally press “SHOW…”, and the form B will appear.

FORM B:

1. In this form you can see and navigate by a loudness graph mixed with the asrun data of the selected channel time span. In the section 15.7 is explained how to read and navigate in loudness graphs.



SHOW module main form – Report by channel-dates

Name	Description
Channel	Indicates the channel that you want to create the loudness report
Date in	Start date of loudness analysis
Date out	End date of loudness analysis
Start offset	Offset in seconds to apply at date in (for debugging purposes)
End offset	Offset in seconds to apply at date out (for debugging purposes)

15.6.5.4 Report by loudness level

This report allows you to search between broadcasted events using integrated loudness (I) criteria, and show the loudness graph of those events.

Follow the instructions to look for loud or quiet broadcasted events:

FORM A:

1. In the main tab section select “By loudness level”.
2. In the “Channel” combobox select the channel you want to analyze.
3. Enter the start time of the time lapse that you want to analyze.
4. Enter the end time of the time lapse that you want to analyze.

LOOK FOR LOUD EVENTS:

5. Enter the integrated loudness threshold in “Search items with loudness higher than”.
6. Press “SEARCH…” in the “High loudness items” section and the form B1 will appear.

LOOK FOR QUIET EVENTS:

5. Enter the integrated loudness threshold in “Search items with loudness lower than”.
6. Press “SEARCH …” in the “Low loudness items” section and the form B2 will appear.

LOOK FOR EVENTS BETWEEN THRESHOLDS:

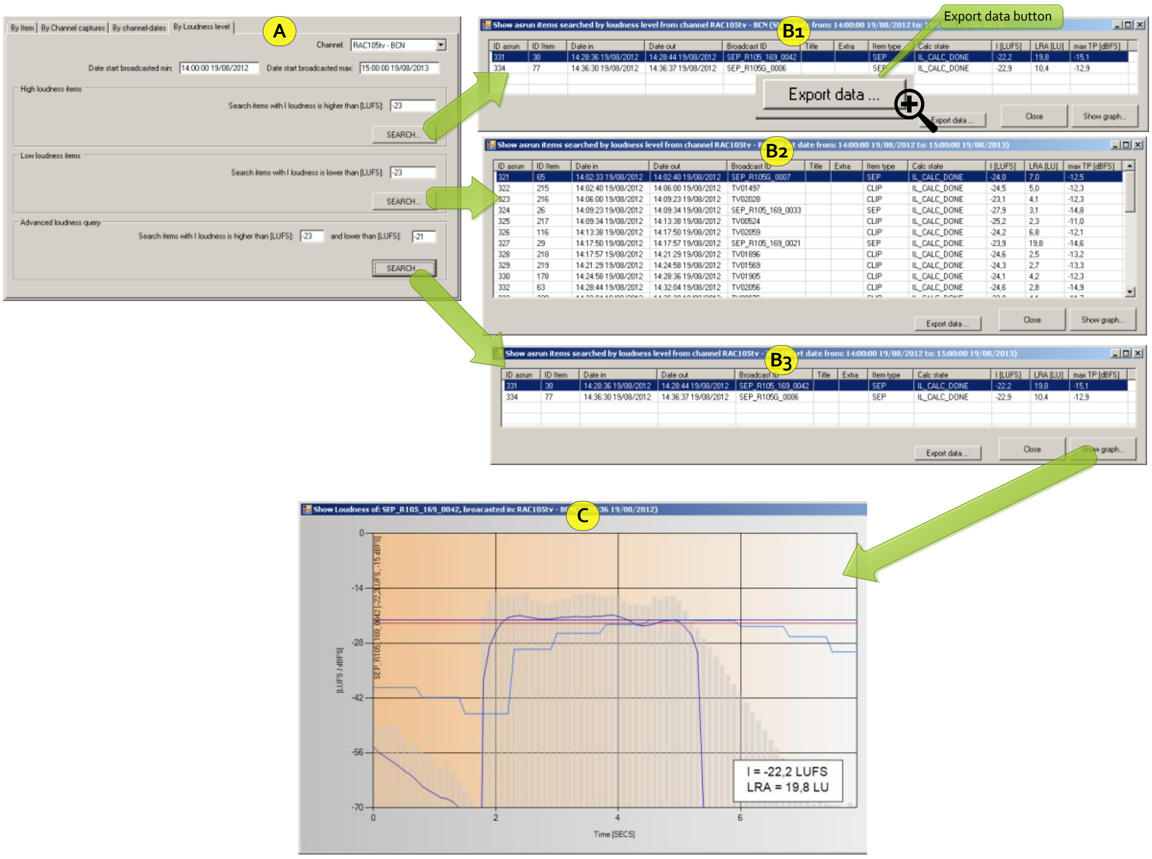
5. Enter the integrated loudness high threshold in “Search items I loudness higher than”.
6. Enter the integrated loudness low threshold in “and lower than”.
7. Press “SEARCH …” in the “Advanced loudness query” section and the form B3 will appear.

FORM BX:

1. In this form you can see a detailed list of broadcasted events that fulfills the search criteria.
2. You can reorder the list by clicking in the list headers.
3. If you can see the loudness graph of one broadcasted event you only have to select it and click “Show graph…”.
4. You can **export data of the report to a text file** by clicking “Export data…” and entering the destination file name.

FORM C:

1. In this form you can see and navigate by a loudness graph of the selected broadcasted event. In the section 15.7 is explained how to read and navigate in loudness graphs.



SHOW module main form – Report by loudness level

Name	***Description***
Channel	Indicates the channel that you want to create the loudness report
Date in	Start date of loudness analysis
Date out	End date of loudness analysis
Start offset	Offset in seconds to apply at date in (for debugging purposes)
End offset	Offset in seconds to apply at date out (for debugging purposes)

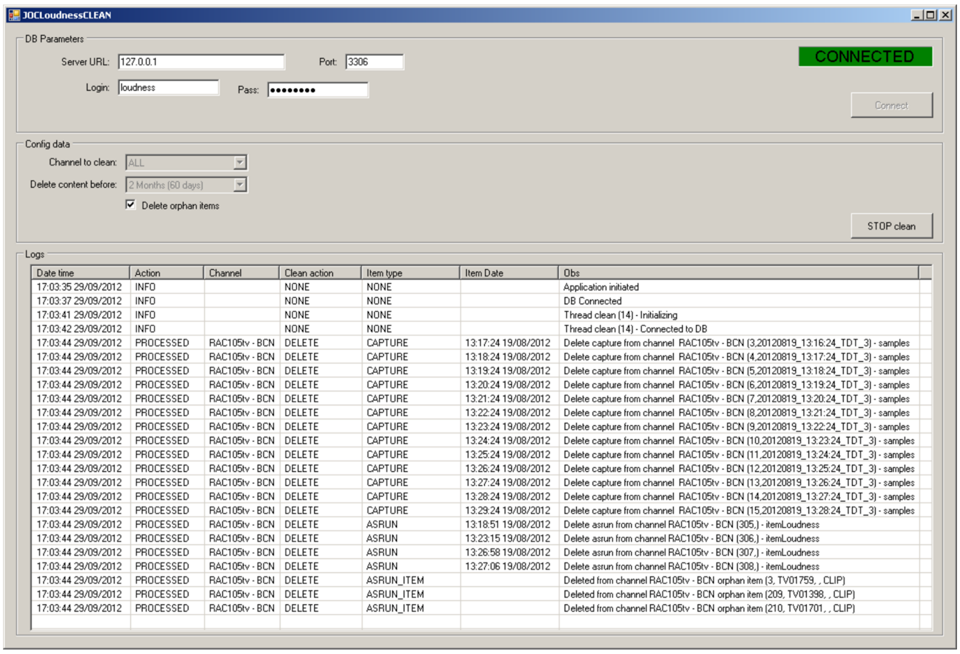
In the following figure we can see a sample of exported report data:

```
#Export loud report
#Channel name
ChannelName = RAC105tv - BCN
#ReportStartTime: Start report time (format HH:MM:SS DD/MM/YYYY)
ReportStartTime = 21:38:55 18/04/2012
#ReportEndTime: End report time (format HH:MM:SS DD/MM/YYYY)
ReportEndTime = 21:38:55 18/04/2013
#Sample order per row (separated by tab) = [idAsrun idItem dteStart dteEnd BroadcastID strTitle strExtra strType CalcState ILoudness LRA maxTP]
Samples = [
1           1           20:13:21 06/04/2013     20:16:52 06/04/2013     TV02234     Closer      Tegan And Sara          CLIP        IL_CALC_DONE            -23,5
            4,1         -13,0
2           2           20:16:52 06/04/2013     20:17:02 06/04/2013     SEP_R105_169_0012       SEPARADOR 16:9 OASIS                SEP         IL_CALC_DONE
            -22,2       5,1         -13,2
3           3           20:17:02 06/04/2013     20:17:29 06/04/2013     AZ000107    PROMO RAC105 OCTUBRE 2012                       PROMO       IL_CALC_DONE
            -24,6       5,2         -11,2
4           4           20:17:29 06/04/2013     20:17:37 06/04/2013     SEP_R105_169_0020       SEPARADOR 16:9 MY LIFE              SEP         IL_CALC_DONE
            -24,9       5,8         -13,9
5           5           20:17:37 06/04/2013     20:20:17 06/04/2013     TV02229     Ho Hey      The Lumineers           CLIP        IL_CALC_DONE            -26,0
            6,6         -12,0
6           6           20:20:17 06/04/2013     20:25:06 06/04/2013     TV00921     The World we Live in    The Killers CLIP        IL_CALC_DONE            -24,1
            3,7         -11,8
7           7           20:25:06 06/04/2013     20:25:17 06/04/2013     SEP_R105_169_0038       SEP 16:9 VOLAR                      SEP         IL_CALC_DONE
            -27,2       19,1        -11,8
8           8
…
];
```

Sample of exported loudness report data

15.6.6 Manual of CLEAN LMS core module

This module deletes automatically the older data of the LMS in order to prevent the system degradation.



GUI of CLEAN module

Name	***Description***
Server URL	Indicates the IP or DNS of the computer where the central DB is installed
Port	Indicates the TCP port that this modules will use to establish the communication channel with central database
Login	The login that this module will use to access to DB
Pass	The password that this module will use to access to DB
Channel to clean	Selects the channel to clean old data (ALL = All channels)
Delete content before	All data behind this threshold will be deleted
Delete orphan items	Indicate if we want to delete items that are not referenced by any asrun entry

To run the clean module you only have to select the channel that do you want to clean (or ALL for al channels), select the period of time of data that do you want to preserve (2 months by default), and click “START clean”.
When the clean module is activated it queries the DB every 1s looking for the old data, and it will delete them if it is found.

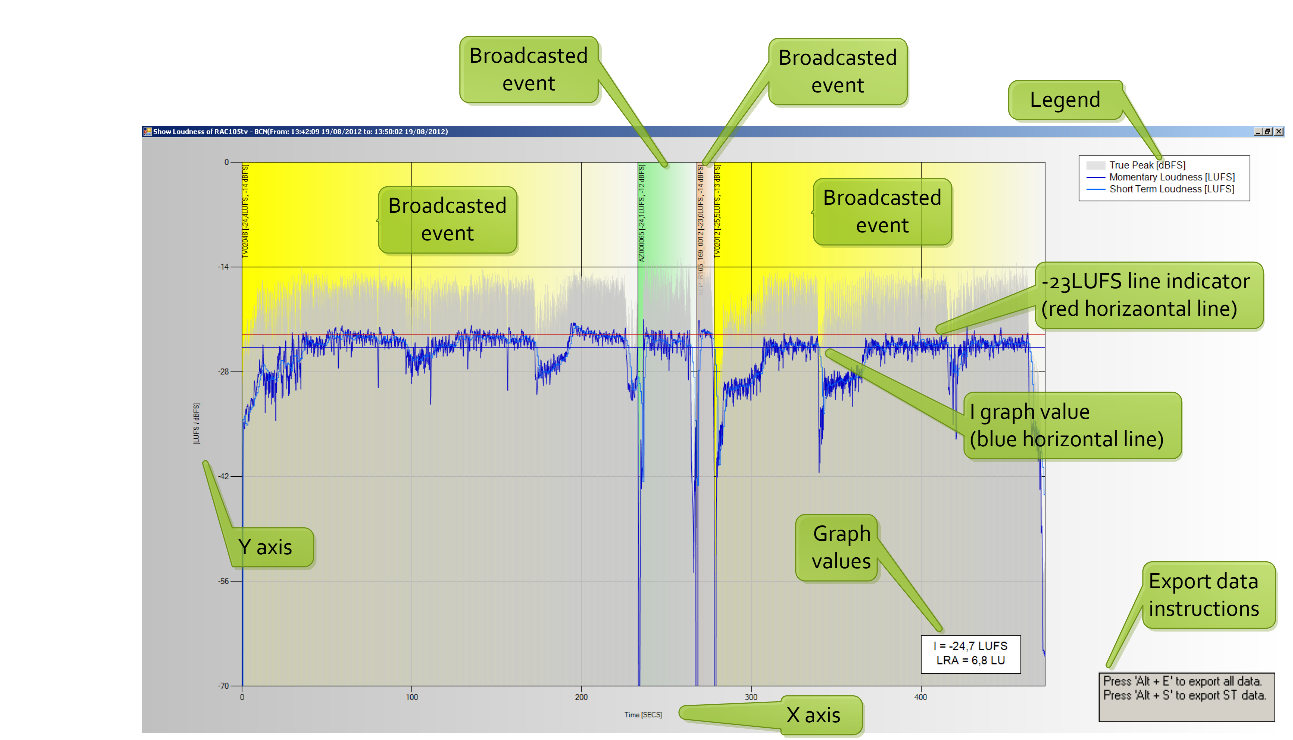
15.7 Loudness graphs

15.7.1 Understanding loudness graphs

The loudness graphs are figures where are showed the following loudness related parameters in function of time:

- Momentary loudness value (M).
- Short term loudness value (ST).
- True peak data (TP).

Depending on the type of loudness graph could appear asrun information in it as well. In the following figure you can see a loudness graph “by dates” report where appears loudness information mixed with asrun information.



Common loudness graph report

The different parts of the loudness graph are:

- X axis (horizontal)
  - Is the time axis and are expressed in seconds.
- Y axis (vertical)
  - The units of this axis are LUFS for loudness values (M, I, and ST), and dBFS for TP measure.
- Graph values
  - Indicates the I and LRA values for all the graph.
- Legend
  - Brief explanation of different measures that are showed in the graph.
- -23 LUFS line indicator
  - Is a horizontal red line that indicates where is the recommended I value in R128.

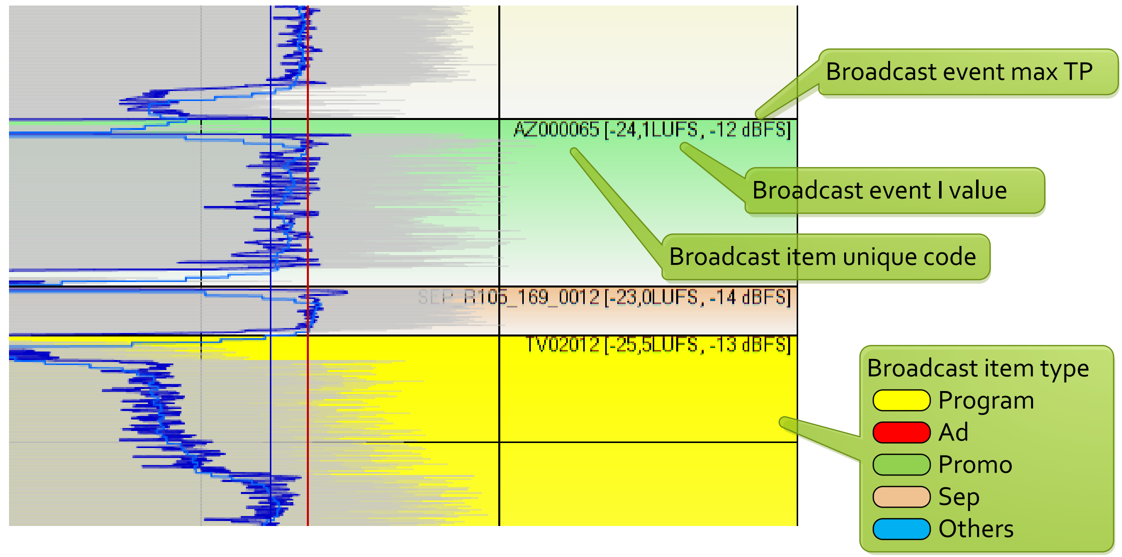
- I graph value
  - Is a horizontal blue line that indicates where is the I value of all graph.

- Broadcasted event
  - Indicates that the colored part of the graph belongs to a single broadcast item. See the about broadcast events in a loudness graphs.

- Export data instructions (see 15.7.4)
  - They are the basic instructions to export the loudness data of the graph to a file. They are only visible during 10 seconds since the graph is loaded.

15.7.2 Broadcasted events in loudness graphs

Every broadcast event has its own loudness information as you can see in the following figure.



Broadcast event in loudness graph

- Broadcast event Maximum True Peak (maxTP)
  - Indicates the maxTP (In dBFS, see 3.2.3) registered in this event.

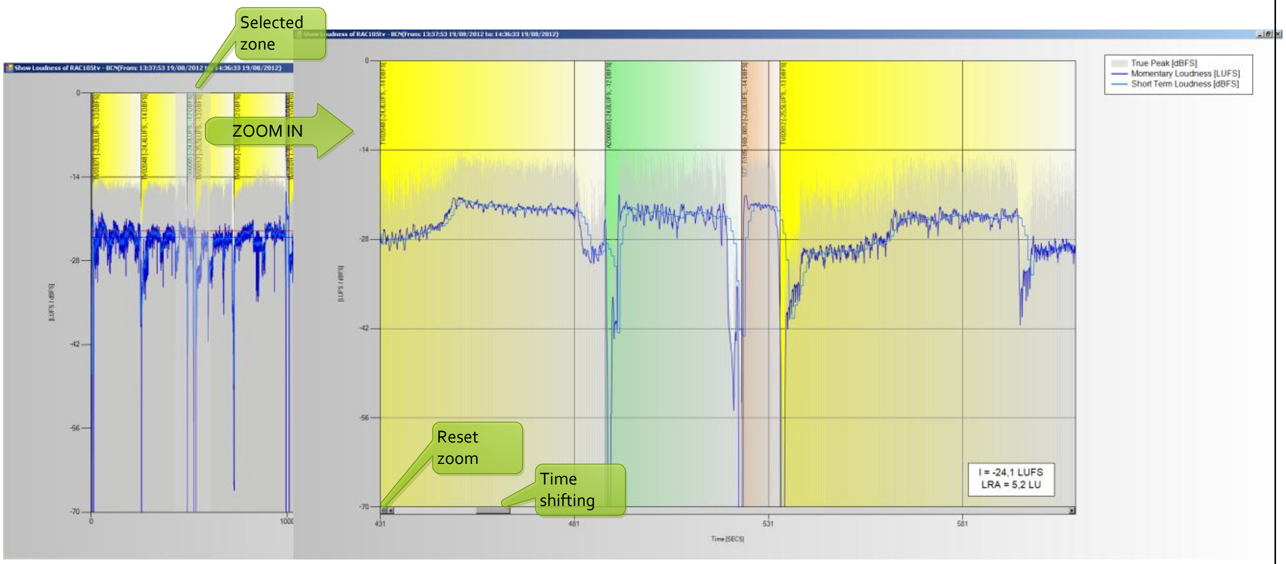
- Broadcast event I value
  - Indicates the integrated loudness (I) value computed for this event.

- Broadcast item unique code
  - Unique alphanumerical code that identifies this event.

- Broadcast item type (color)
  - The background color of the event identifies the type of itself.

15.7.3 Zooming in loudness graphs

You can navigate inside loudness graphs zooming into the interest zones. To do that you only have to select (left click pressed) to the graph zone that you want to zoom in, see the following figure.



Zoom in loudness graphs

Once the zone of your interest is zoomed you can:

- Reset the zoom and goto to the previous view by clicking left-bottom button.
- Shift the graph using bottom slider.

15.7.4 Export loudness graph data

To export the loudness data to a text file you only have to select the graph windows that you want to export their data and press the following key combination depending on the data that you want to export:

- Alt + e: To export all graph data, that includes:
  - Clock data: Timming data, the amount of milliseconds since capture started
  - Loudness momentary values (M)
  - True peak values (TP)
  - Loudness short term values (ST)
- Alt + s: To export only the ST data of the graph,
  - Clock data: Timming data, the amount of milliseconds since capture started
  - Loudness short term values (ST)

At this point you will have to enter the destination file, and the data will be exported. In the following figure you can see a sample of an exported data file.

```
#Export loud data
#Channel name
ChannelName = TV3 - BCN
#Fs: Sampling frecuency
Fs = 48000
#SampleLoudInterval: The interval between 2 loud samples (in audio samples)
SampleLoudInterval = 4800
#SampleLoudIntervalMS: The interval between 2 loud samples (in milliseconds)
SampleLoudIntervalMS = 100
#StartTime: The time when the capture starts (format HH:MM:SS DD/MM/YYYY)
StartTime = 20:11:50 07/04/2013
#Sample order per row [linear sample] = [clk M TP ST]
#Remember MinLUFS =  10.0 * log10(M) - 0.691
#Remember TLindBFS = 20.0 * log10(dVal)
Samples = [
11 4,02324710314262E-05 0,0294256656990979 0,00690717010438646
73 4,36902615114122E-05 0,0224429136179399 0,00690717010438646
222 4,67688177027913E-05 0,0171171784799335 0,00690717010438646
261 4,98015016397859E-05 0,0395363317535324 0,00690717010438646
409 6,49148793579984E-05 0,0301543043201891 0,00690717010438646
…
450 7,87380257288536E-05 0,041756754655781 0,00690717010438646
];
```

Sample of exported loudness graph data